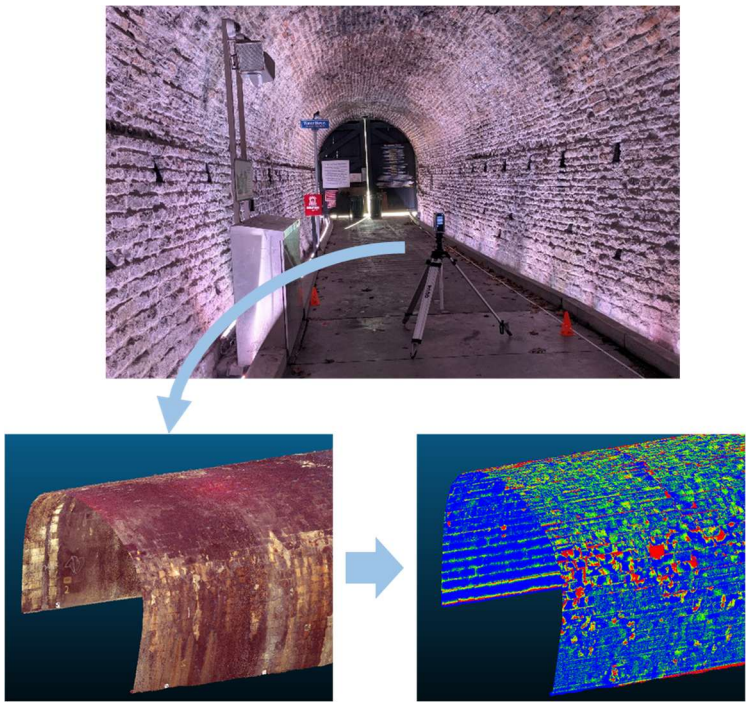
AI-Powered Damage Segmentation for Automated Condition Assessment of Historic Masonry-Lined Tunnels

Jack Martin William Smith

University of Leeds

School of Earth and Environment



Thesis submitted in accordance with the requirements for the degree of

Doctor of Philosophy

October 2025

Intellectual property statements

I confirm that the work submitted is my own, except where work which has formed part of jointly authored publications has been included. My contribution and the other authors to this work has been explicitly indicated below. I confirm that appropriate credit has been given within the thesis where reference has been made to the work of others.

Publication 1

Smith, J., Paraskevopoulou, C., Cohn, A.G., Kromer, R., Bedi, A., Invernici, M.,. 2024. Automated masonry spalling severity segmentation in historic railway tunnels using deep learning and a block face plane fitting approach. *Tunnelling and Underground Space Technology 153, 106043*. https://doi.org/10.1016/j.tust.2024.106043

- Chapter 5 of this thesis explains the automated workflow proposed in this publication.

- Chapters 6, 7 and 8 expand upon work developing neural networks for masonry joint segmentation first published here.

- Chapter 9 includes analysis conducted in this publication

Publication 2

Smith, J., Paraskevopoulou, C.,. 3D Deep Learning for Segmentation of Masonry Tunnel Joints. Structural Integrity Procedia, proceedings of the 7th International Conference on Smart Monitoring, Assessment and

Rehabilitation of Civil Structures, SMAR 2024, Volume 64, Pages 220-227. https://doi.org/10.1016/j.prostr.2024.09.233

-Chapter 6 re-evaluates and expands upon work using 3D neural networks and the XGBoost algorithm for masonry joint semantic segmentation first proposed in this publication.

Publication 3

Smith, J., Paraskevopoulou, C.,. 2025. Practical assessment of masonry tunnel joint segmentation using topological machine learning. *Civil Engineering Design 7, 93–110.* https://doi.org/10.1002/CEND.202400049

- Chapter 7 presents work from this publication

Publication 4

Smith, J., Paraskevopoulou, C.,. 2025. Uncertainty Quantification to Assess the Generalisability of Automated Masonry Joint Segmentation Methods. *Infrastructures (Basel) 10, 98.* https://doi.org/10.3390/INFRASTRUCTURES10040098

- Chapter 8 presents work from this publication

For each publication listed, my contribution was as follows:

- Literature search
- Development of methodology
- Data collection, unless otherwise referenced
- Development of research code, unless otherwise referenced
- Data analysis
- Figure creation
- Drafting of manuscript

For each publication listed, my co-authors contributed as follows:

- Project conceptualisation
- Obtained funding
- Provided overall advice and supervision
- Manuscript review and editing support

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. I would like to thank EPSRC and the University of Leeds for giving me the opportunity to conduct this project and providing the resources for me to learn effectively.

I would like to especially thank my supervisor, Chrysothemis Paraskevopoulou for inviting me onto this journey and generally providing so much support and advice throughout the four years. Thank you for encouraging me to attend so many conferences, for helping me network and always championing my work.

I would like to also thank my co-supervisors Anthony Cohn and Ryan Kromer for providing great discussions about my work.

I would like to acknowledge Bedi Consulting Ltd. for providing important operational railway tunnel data for the project. Thank you Anmol Bedi and Marco Invernici for providing much practical support and advice.

A big thank you to Mark Diederichs from Queen's University for inviting me to Kingston and guiding me through many hours of data collection in the Brockville tunnel.

Thank you also to everyone in the Scientific Machine Learning group at Leeds, who introduced me to many new techniques, helped me through coding nightmares and most of all provided a really friendly community - in addition to keeping me well fed with plenty of coffee and cake.

Finally, I would like to reserve a special thanks for my partner, family and friends who have all continuously supported me and put up with many conversations about bricks and tunnels over the years!

ABSTRACT

Historic masonry tunnels form a substantial part of the underground railway infrastructure globally. To ensure their stability, regular condition inspections must be undertaken that typically involve a lengthy and subjective manual defect labelling process. Surface damages can be extensive on the lining of historic masonry tunnels. These are the multiple different types of surface damage such as spalling, efflorescence, degraded mortar and biological growth. This thesis investigates the potential of automated methods to improve the manual procedure typically employed within industry for masonry lined tunnel condition assessment.

Masonry spalling severity, defined by the depth of spalling, is a key indicator of a masonry tunnel's condition. This study proposes an automated workflow for identifying spalling severity from 3D tunnel lining point cloud data obtained by lidar. The workflow combines using deep learning for masonry joint and damage semantic segmentation with a geometric undamaged masonry plane fitting procedure to recreate the tunnel lining without surface damages. The study first uses synthetic data to investigate the workflow's limitations before analysing the most effective algorithms for masonry joint and damage segmentation. Supervised convolutional neural networks were selected for masonry joint and damage segmentation. Training with topological loss functions and a transformer-based encoder were shown to improve performance. Using masonry joint segmentation as a proxy for the automated workflow's performance, possible solutions to the challenges of generalising the method between different tunnels, quantifying uncertainty and determining the optimal training method are presented. Training a method on different tunnels alongside targeted tunnel specific training is shown to achieve the best performance.

The study finally evaluates the trained spalling severity segmentation algorithm on multiple real-world tunnels, demonstrating effective masonry spalling localisation. The workflow's robustness enables it to always provide a useful indication of potentially damaged areas that a human assessor can then analyse in more depth. The presented work shows how an automated method using deep learning can be integrated into routine masonry tunnel condition assessments to reduce analysis time and generate more comprehensive spalling maps.

Table of Contents

List of Figures

List of Tables

Abbreviations

2D	Two Dimensional
3D	Three Dimensional
AVC	Area Variation Coefficient
BIM	Building information Modelling
BWIoU	Blockwise Intersection over Union
CIDice	Centreline Dice score
CNN	Convolutional Neural Network
DPT	Damage Proportion Threshold
GPR	Ground Penetrating Radar
IoU	Intersection over Union
KPConv	Kernel Point Convolution
Lidar	Light Detection And Ranging
MCD	Monte Carlo Dropout
MLS	Mobile Laser Scanning
NN	Neural Network
Radar	Radio Detection And Ranging
R-CNN	Region-based Convolutional Neural Network
SLAM	Simultaneous Location And Mapping
TLS	Terrestrial Laser Scanning
TTA	Test-Time Augmentation
UMIOU	Uncertainty Mean Intersection over Union
XGBoost	Extreme Gradient Boosting

1. Introduction and Outline

1.1. Problem statement

Rapid economic development during the second half of the 19th century saw extensive expansion of railway infrastructure and the construction of most railway tunnels in use today both in the UK and worldwide. These tunnels predominantly utilise arches of stone or brick masonry. As a result, continued safe operation of these historic structures is vital for maintaining a reliable railway network and protecting a valuable section of cultural heritage (Atkinson et al., 2021; Keshmiry et al., 2024); especially given the increasing demand for more efficient and cost-effective lower-carbon transportation options. It is substantially cheaper and more environmentally considerate to enhance repair and maintenance strategies to extend a tunnel's structural lifespan (McKibbins et al., 2010; Neves et al., 2018; Richards, 1998) than to build new, with maintenance work typically forming a small proportion of a tunnel's life cycle cost (Jadhav and Nichat, 2021; Petroutsatou et al., 2021) and initial construction the greatest cost (Membah and Asa, 2015). Furthermore, large scale construction and rehabilitation works requiring tunnel closures within usual working hours are typically substantially more disruptive to tunnel users (Papakonstantinou et al., 2025; Tympakianaki et al., 2018) than targeted small scale maintenance work that can take place overnight. However, conducting efficient maintenance requires accurate documentation of a structure's geometry and condition (Neale, n.d.; Shou et al., 2014).

Inspections of tunnels typically consist of teams of assessors walking through a structure and noting down by hand the location and extent of any defects

(Chiu et al., 2015; Llanca et al., 2017). Within the UK, inspections of masonry lined tunnels are conducted using broadly similar standardised procedures (Network Rail, 2023). Despite this, it is a laborious task subjective to the inspecting engineer's judgement (Forster and Douglas, 2010), with the level of detail restricted by the inspectors' time on site. Damages such as lining cracking and brick loss are often localised and easy to visually identify onsite on historical tunnel linings. However, spalling damage is usually much more widespread (Chiu et al., 2015), although most is of low severity. As a result, Identifying the location and severity of masonry spalling damage is a particularly time consuming procedure that limits the attention that an assessor can give to more critical damages such as lining cracking and deformations that more directly impact the structural integrity and stability. It is therefore challenging to reliably conduct further structural modelling and analysis from these reports.

1.2. Justification and Aims

There have been substantial recent advancements in machine learning (Jordan and Mitchell, 2015; Spencer et al., 2019; Voulodimos et al., 2018), robotics (Lattanzi et al., 2017; Montero et al., 2015) and other automation technologies (Agnisarman et al., 2019a; Newman, 1995; Son et al., 2015). With the successful application of many of these emerging technologies to the condition assessment of concrete infrastructure (Halder and Afsari, 2023; Koch et al., 2015; Omar and Nehdi, 2018; Sjölander et al., 2023), it is likely that these methods will also be able to improve the condition assessment procedure of historic masonry lined tunnels. By bringing masonry lined tunnel condition assessment techniques in line with those developed for modern structures (Kennedy et al., 2024)(Kennedy et al., 2024), their condition can be more accurately characterised and more efficient, targeted remediation work can be conducted. This study aims to develop an automated method for analysing a masonry tunnel's lining using routinely collected data.

It was chosen to focus analysis on the identification and localisation of masonry spalling damage. The typically large extent of spalling on the surface of a masonry tunnel's lining amplifies the labour-saving impact of automating spalling assessment compared to the detection of other damages. Furthermore, there should be enough examples of masonry spalling on a typical tunnel's lining to effectively train a machine learning model should a supervised method be used. The defined depths of spalling severity categories makes it possible to manually benchmark any developed automated method. As a result, the study focuses on developing automated methods for creating spalling severity maps and aims to achieve the following key objectives:

1. Develop an automated method of masonry tunnel lining spalling assessment that is compatible with existing tunnel condition assessment standards and can reduce the labour required to conduct a masonry tunnel condition assessment.
2. Demonstrate the workflow's effectiveness on a variety of real masonry tunnel lining situations and provide a comparison against existing spalling detection methods.
3. Show how an automated workflow can be integrated into the typical industry condition assessment procedure and how it can aid a human tunnel assessor.
4. Examine the variability in the output of deep learning-based methods when applied to different masonry lined tunnels in order to understand the workflow's reliability when applied in practice.
5. Propose a framework for understanding the limitations of a proposed automated analysis method to enable tunnel assessors to make an informed decision on whether the method will be suitable for a specific tunnel.

1.3. Scope of research

This thesis proposes an automated workflow for identifying tunnel masonry spalling damage from 3D point clouds of a tunnel's lining obtained by lidar. The workflow operates as follows. 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 machine learning model is used to segment masonry joint locations to isolate individual blocks. A separate machine learning model 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.

In order to meet the study's objectives, the performance of the machine learning aspects of the workflow are trained and evaluated on data from four different tunnels from the UK that were made available for this study. A fifth tunnel, the Brockville tunnel in Canada, was surveyed as part of the project and it's dataset was reserved for a final real world performance evaluation of the workflow. Performance of the workflow will be assessed against whether it improves outcomes for any of the following factors:

- Health and safety.
- Cost
- Analysis time
- Time required on site
- Defect identification accuracy
- Assessment reproducibility
- Communication of assessment results

1.4. Outline

Chapter 1 - Introduction and Outline

This chapter briefly introduces the project's background. It explains the justification for the project and presents its key objectives. It then provides an outline of the project's research approach in each chapter.

Chapter 2 - Masonry Lined Tunnels

Focusing on a UK context, this chapter provides a historical overview of the development of masonry lined tunnel infrastructure. It explains different masonry lining construction methods and what factors influence the lining's stability. The chapter explains the causes of commonly occurring damages before outlining how these damages are assessed today. The chapter concludes by specifying the most important factors limiting the performance of masonry lined tunnel condition assessments.

Chapter 3 - Automated Condition assessment: an overview

In order to select the most promising technologies to further develop to aid with masonry lined tunnel condition assessments, this chapter provides an extensive review of methods that have been developed for the assessment of all types of tunnels. It first evaluates the different available data collection techniques, before investigating automated approaches for tunnel data analysis focusing on semantic segmentation of tunnel lining surface damages. The chapter finally presents research recommendations for this study, noting which areas are under researched in the literature.

Chapter 4 - Available datasets and resources

This chapter introduces each of the tunnel 3D point cloud datasets made available for this study. It then describes the data preprocessing steps which were undertaken and fall outside the scope of the automated methods presented in later chapters. The chapter also outlines the computing systems utilized during the study for running developed code and analyses.

Chapter 5 - A proposed workflow for masonry tunnel condition assessment

Each step of the proposed workflow for automated masonry lined tunnel spalling severity segmentation is described in detail in this chapter. Using synthetically generated masonry lined tunnel point clouds, the outputs of each step of the workflow are visualized. An analysis of the sensitivity of the workflow's overall performance to the accuracy of the joint and damage semantic segmentation steps is also conducted.

Chapter 6 - Computer vision for masonry joint and damage segmentation

This chapter compares the most promising methods for conducting the masonry joint and damage segmentation steps of the proposed workflow. Using masonry joint segmentation performance as a guide, it compares traditional computer vision methods with decision tree, 2D and 3D Convolutional Neural Network (CNN) approaches. The chapter conducts an extensive search for the most effective method and hyperparameter combination and presents the best performing segmentation method for each task.

Chapter 7 - Generalization analysis

This chapter investigates how joint segmentation performance can be improved using topological loss functions and assesses how the 2D CNN methods perform under the different training regimes that may be available when preparing the workflow for real world tunnel condition assessments. With the context of low training data availability, the chapter shows how much performance can be expected to vary between different trained joint segmentation CNNs.

Chapter 8 - Performance prediction

Quantification of a machine learning method's uncertainty enables a tunnel assessor to predict the effectiveness of the method and determine which areas should be manually checked. This chapter evaluates two methods for quantifying the uncertainty of a CNN trained for masonry joint segmentation.

Chapter 9 - Workflow validation and discussion

This chapter evaluates the performance of the complete proposed spalling severity segmentation workflow using the 2D CNN based masonry joint and damage semantic segmentation methods selected in Chapter 6. It provides a quantitative and qualitative assessment of spalling severity segmentation performance on an example brick and stone lined tunnel. The chapter then conducts an in-depth analysis of the workflow's real-world effectiveness when applied directly to the Brockville tunnel without tunnel specific training or tuning. The output of the workflow is compared against that of a change detection analysis and a manual surface damage assessment, where it is shown to be faster and more reliable than the manual method.

Chapter 10 - Conclusion

The thesis' key outputs are discussed and the impact of the research is presented. Recommendations are made for future studies.

2. Masonry lined tunnels

2.1. Historic background

The opening of the world's first inter-city railway from Manchester to Liverpool in England in 1830 set off a transportation revolution that led to railways becoming the preferred method of long distance overland transportation for much of the next century. The speed, safety, labour efficiency and capacity of railways surpassed that of canals and horse drawn coaches for both goods and passenger transportation Simmons, 1991). The industrial revolution brought increased urbanisation and growing demand for the transport of coal and manufactured goods to urban centres. New railways both benefitted and fuelled this growth, leading to an exponential expansion of railway mileage in England and Wales. By 1870, there were 15,537 miles of railway in the UK and Ireland (*The Railway Year Book for 1915*, 1915). Alongside, a substantial number of new tunnels and bridges were built to accommodate the new railway infrastructure.

Early rail tunnels were constructed with unprecedented urgency and required much larger diameters than the canal tunnels that preceded them. Constructed before the advent of modern design standards or analysis methods, construction was costly, labour intensive and dangerous. Nevertheless, multiple drivers led to extensive construction of new tunnels:

- Early steam locomotives could not handle sharp gradients (Tyler, 1867) , so tunnels were built to enable railways to travel through mountainous terrain while avoiding excessive inclines.
- Strong competition between railway companies led them to seek faster routes, away from traditional ones with more favourable

topography. This can be observed in the multiple routes constructed through the Pennine Hills (Casson, 2009).

- The need to reach a convenient location in existing urban centres resulted in new tunnels being constructed underneath built-up areas to avoid complicated and controversial land purchases and building demolitions. One of the first urban railway tunnels, the Waterloo Tunnel (Perry, 2012), was completed in 1849 by the Liverpool & Manchester railway, connecting their main line directly into the centre of Liverpool.
- Often, rural landowners would object to new railways passing through or splitting up their land. Sometimes an agreement would be reached where an accommodation tunnel would be constructed to shield the railway from view. An example of this is the Midland Railway's tunnel under the Haddon estate (Waite and Harris, 2012).

The combination of these factors in the mid 19th century resulted in the construction of over 340km of tunnel that now forms the majority of the 622 railway tunnels operational in the UK today (railwaydata.co.uk, 2024)(.

Within the UK, railway infrastructure milage peaked in the 1930s, before declining in the post World War II period (Gibbons et al., 2024)(due to the growth of road transport and decline of heavy industry within the UK. As a result, very few new mainline railway tunnels were built in the UK in the 20th century. A related pattern of growth and decline is observed in other countries that industrialised rapidly in the 19th century, with a more extreme decline observed within the US (Stover and Carnes, 1999) and less of a decline observed in continental Europe (Martí-Henneberg, 2021).

British railway passenger numbers have rebounded since the late 1990s (Coucher et al., 2008). However, recent efforts to modernise and expand the

British railway network have been slow (Stewart, 2025) and a large proportion of the in-use railway infrastructure in the UK is now over 100 years old. Upcoming net zero targets necessitate expansion of railway capacity to enable further modal shift of both passengers and freight traffic to rail (Klaaßen and Steffen, 2023). This requires either the expansion of existing tunnels (Atkinson et al., 2021), or the construction of costly new tunnels (Cross, 2018). Despite this, with increasing cost pressures on governmental infrastructure spending, investment in infrastructure enhancements has reduced in the UK (Office of Rail (UK), 2023). As a result, there is a substantial pressure to more efficiently utilise the existing historic infrastructure. Overall, aging railway infrastructure is an issue that is being faced globally and, given the high cost and uncertainty that has been associated with new tunnel construction (Paraskevopoulou and Benardos, 2013; Paraskevopoulou and Boutsis, 2020), it is paramount that the lifespan of existing tunnels can be extended for railways to remain economical.

2.1.1. Victorian tunnels

Most railway tunnels constructed in the 19th century were lined with masonry. Stone and brick masonry was the preferred material for lining tunnels from ancient times through to the early 20th century. The first reference to a masonry lined tunnel was by Diodorus of Sicily in the 1st Century BCE who mentioned a legendary ancient Babylonian tunnel under the Euphrates River. By the time of the arrival of the railways, the construction of masonry structures was routine, and experience of underground works had already been built up through the first industrial revolution from the construction of sewers, aqueducts and canals in the 18th century. However, understanding of rock and soil mechanics was still limited and most construction decisions were based on an engineer's personal judgement and rules of thumb formed

from previous successful constructions. Later engineers continued to use these rules through reference books such as Molesworth's Pocketbook (Molesworth, 1886).

In general, early tunnels were constructed by either boring through good rock or using a cut and cover technique to create a shallow tunnel (Drinker, 1882; ROGERS, 1901). The cut and cover method involves first digging a cutting, then constructing a masonry arch above and infilling on top with rubble to regenerate the ground surface. Once the cutting has been created, construction is effectively the same as that of an elongated masonry arch bridge. Although the construction of cut and cover tunnels is limited to locations where the route is not obstructed above ground, it was a popular method of metro railway construction and was used for the Metropolitan line in London and much of the Paris metro where tunnels could be built under existing roads.

Where cut and cover was not possible or a deeper tunnel was required, engineers first considered the underlying geology. Victorian engineers had limited knowledge of rock behaviour and there was usually very little ground investigation – construction would be adapted onsite when different conditions were encountered. Ground was broadly classified into being either Light and largely self-supporting or Heavy, which requires substantial support due to higher ground pressures. (Gripper, 1879) suggests that unless the tunnel is within a strong geological formation, then heavy ground should be expected below 50ft (~15m) depth. The reactive construction method combined with a lack of knowledge of long-term ground behaviour has often led to issues that still manifest today where there was a transition between lining type.

The digging of early Victorian tunnels was conducted exclusively with hand tools. Although this was labour intensive and the work of the navvies (railway construction workers) was often very dangerous (Coleman, 1981), progress was relatively swift. For example, the 2.2km Kilsby Ridge tunnel on the London-Birmingham railway was constructed in the 1830s in just 3 years. It required 1250 navvies, 26 of whom died during the construction. Despite issues with flooding, rapid progress was achieved by first constructing 18 vertical shafts from the surface along the length of the tunnel, enabling multiple faces to be excavated simultaneously (Beaver, 1972). The use of multiple access shafts was usual for the construction of deep tunnels. Later tunnels used a ‘break-up’ method whereby a narrow pioneer tunnel was first driven between shafts. Construction on the main tunnel was then broken up from multiple points along the pioneer tunnel, enabling even more simultaneous excavation. After the invention of dynamite in 1864, explosives were used in harder ground, however they were difficult to control and led to an increased amount of unplanned overbreak.

In cases where the ground was not self-supporting, temporary supports were required before the lining could be constructed. Many countries had their own preferred method of excavation staging. The English method involved excavating down from the crown and sliding forward temporary timber formwork as different levels were reached. A key aspect of this method is that the masonry lining was not constructed until after the entire section had been excavated (Beaver, 1972). The advantage of this is that the masonry sidewalls and arch are constructed from the bottom up, resulting in a higher quality lining within each excavated section. However, this led to a greater time delay between excavation and lining construction during which the ground may converge (Paraskevopoulou and Diederichs, 2018), reducing

the diameter of the opening and preventing the desired profile from being achieved. This also often led to a poor connection being made at the joint between the masonry of different sections.

Tunnelling in softer ground was not possible until the invention of the first tunnelling shields. Perfected by Greathead in 1869 during the construction of the Tower subway in London, the tunnel shield enabled better protection of work at the face. In addition, by sliding forward the same cast iron shield, a more consistent lining shape could be achieved than that possible with timbers. Later innovations using compressed air at the face to prevent water ingress enabled tunnelling through water-bearing strata (Copperthwaite, 1906). However, masonry was poorly suited to the force exerted by the hydraulic jacks used to push the shield forwards, so a cast iron lining was preferred.

Cast iron began to replace masonry for lining tunnels in the late 19th century primarily due to its strength. Since a cast iron lining could be made thinner than a masonry one, less excavation was required to obtain the same interior diameter. With labour becoming more expensive, masonry tunnel construction became substantially more costly than cast iron, and later steel reinforced concrete (Groves, 1943). From an engineer's perspective, it was also easier to modify and predict the behaviour of cast iron than masonry.

Modern railway tunnels generally have concrete or shotcrete linings and masonry is mostly used for aesthetic purposes in new railway infrastructure. The predominant demand for masonry construction expertise is in the repair of existing infrastructure. Nevertheless, where railway lines have been kept open, existing masonry tunnels have largely endured with relatively limited maintenance to the present day and now form an important part of our industrial heritage. The conservative rules of thumb used in 19th century

tunnel design resulted in overbuilt tunnels and the robustness and repairability of their masonry linings is what has enabled many Victorian era tunnels to continue being used today.

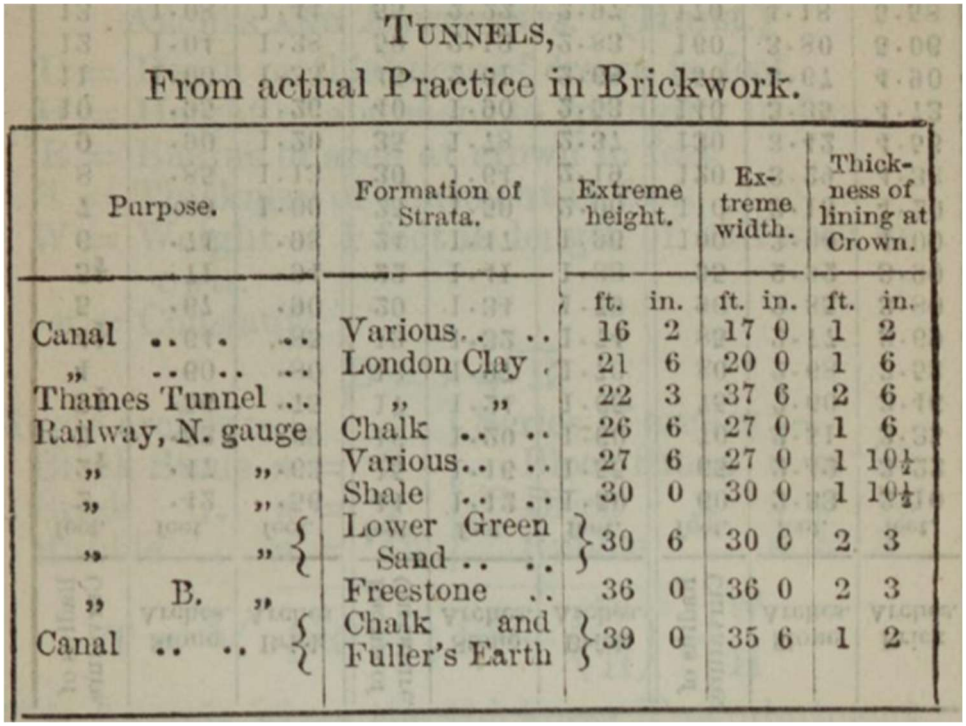
2.1.2. Masonry lining

The primary role of the tunnel lining is to resist the ground pressure and retain the tunnel profile. In self-supporting ground, a lining may still be provided for protection from water ingress and loose rocks. Where a tunnel passes through a succession of geologies with varying properties, an engineer may choose to either build a thicker lining throughout to account for the worst case location, or regularly vary the design as different geologies are encountered. The engineer must balance the cost of providing a thicker lining with the difficulties of constructing multiple joints at lining type transitions.

Within the UK, masonry tunnel linings were typically constructed from locally produced brick, although in some locations stone was used where it was readily available. The construction of stone masonry was in general slower than that of brick, as larger blocks needed to be transported and cut to size on site. Early railway tunnel linings were typically constructed as an arch on vertical sidewalls, although later designs used a horseshoe or oval profile to better resist the lateral ground pressure. Often, as built drawings were either not made or have since been lost, so it can be challenging to determine a tunnels design profile from subsequent lining deformations.

The number of layers of masonry would be varied depending on the desired arch strength. Commonly used rules of thumb for linings in different stratigraphies are shown in Figure 2.1. Both stone and brick masonry are strong in compression but have practically no tensile strength. As a result, an arch must be designed so that the thrust line passes around the arch and

keeps it in compression. If the thrust line passes outside the arch, then a pivot point will form, and the lining will become unstable and may collapse. The arch must be designed to accommodate long term convergence of the surrounding ground post construction and be robust to future changes in the location of the thrust line.



TUNNELS,
From actual Practice in Brickwork.

Purpose.	Formation of Strata.	Extreme height.	Extreme width.	Thickness of lining at Crown.
		ft. in.	ft. in.	ft. in.
Canal	Various.. ..	16 2	17 0	1 2
„	London Clay .	21 6	20 0	1 6
Thames Tunnel ..	„ „	22 3	37 6	2 6
Railway, N. gauge	Chalk	26 6	27 0	1 6
„ „	Various.. ..	27 6	27 0	1 10½
„ „	Shale	30 0	30 0	1 10½
„ „	Lower Green Sand	30 6	30 0	2 3
„ B. „	Freestone ..	36 0	36 0	2 3
Canal	Chalk and Fuller's Earth	39 0	35 6	1 2

Figure 2.1: 19th century rules of thumb for determining lining thickness, from Molesworth (1886)

Various bond patterns have been used in masonry tunnels. The most common consisted of a varying number of stretcher layers between single header layers. Stretcher layers have masonry blocks laid in line with the tunnel axis, while header layers are laid radially. This is shown in Figure 2.2. The English bond was popular due to its familiarity from use in buildings and consists of a single stretcher layer between header layers. However, as shown in Figure 2.2, the curvature of the tunnel arch leads to a greater gap between the bricks on the outside of the lining, concentrating stress towards the inner lining. It is preferable to tailor the number of

stretcher layers based on the curvature of the arch, a method known as a running bond. However, sometimes the English bond was preferred due to the better bonding between the layers preventing ring separation (Gilbert and Melbourne, 1995). After the construction of each section of lining, any gaps between the masonry and surrounding ground would be infilled with rubble to ensure an even loading on the masonry arch.

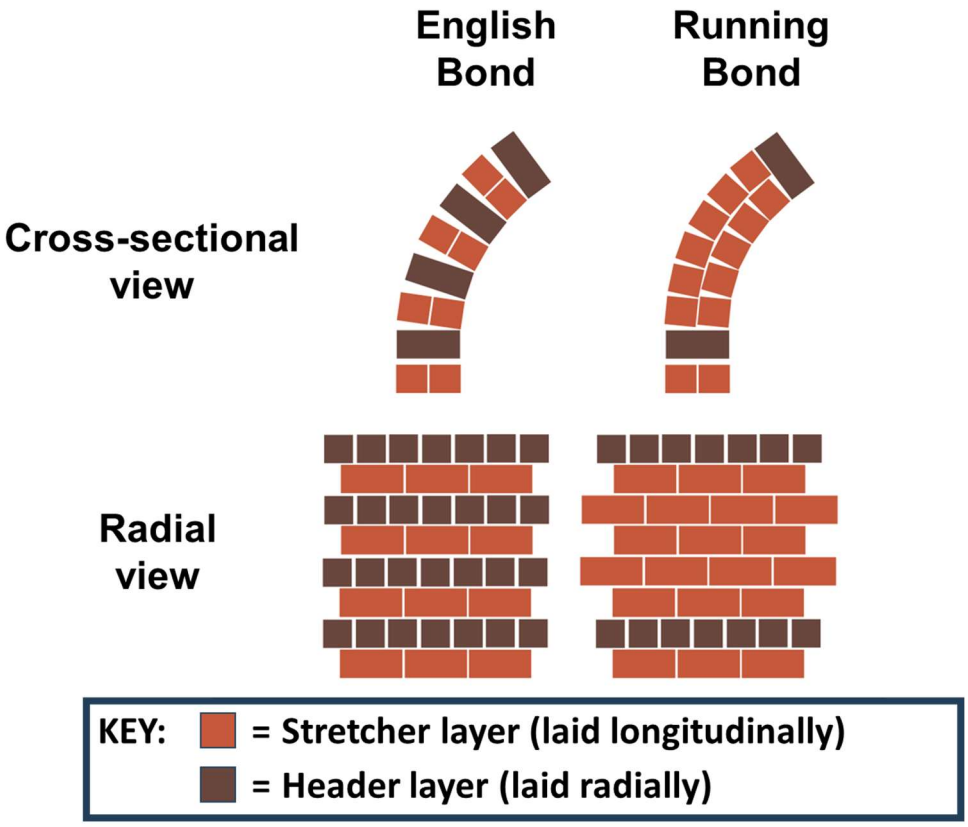


Figure 2.2: Two masonry bonding patterns commonly used in 19th century tunnels. The figure shows how adapting the frequency of header layers to match the arch curvature reduces the gap between blocks on the outside of the arch.

2.2. Masonry tunnels today

After over 100 years of use, the linings of many Victorian era tunnels have deteriorated, and serviceability has been largely maintained by frequent masonry patch repairs. Tunnel collapse will occur when the lining is either overloaded or the thrust line of the arch moves leading to instability (Heyman, 2014)(Heyman, 2014). These can either be caused by changes in the loading state or deterioration of the properties of the arch. Reduction in the effective thickness of the lining due to brick loss will cause overloading and voids forming behind the lining is the most common cause of instability.

While some tunnels may be damaged by one time events such as train derailments, vandalism or earthquakes, most deteriorations are caused by long term repeated exposure to adverse conditions. A summary of the most common deterioration drivers is shown in Table 2.1. However, many issues currently impacting the lining serviceability are exacerbated by errors made during construction. Common construction flaws are outlined in Table 2.2. In addition to these, many tunnels have been subject to patch repairs made from materials that are not sympathetic with the original lining. Differences in the chemical composition of newer mortar can accelerate chemical attack and prevent an effective bond being formed between new and original lining areas. Differences between the elastic modulus of original and repaired areas can exacerbate changes in the loading state leading to greater stress concentrations.

Table 2.1: Key drivers of masonry lining deterioration

Deterioration driver	Explanation	Long term Impact
Water ingress	Water can enter the tunnel through cracks in the lining and due to the porosity of the mortar used in brick linings. This can wash out fill and looser areas of masonry and mortar.	Detachment of masonry, loss of mortar
Salt deposits	Salts suspended in water permeating through the masonry can precipitate, breaking down the structure of the masonry.	Masonry spalling, loss of mortar
Chemical weathering	Ground water can be acidic or contain sulphates. These will cause the breakdown of bricks and stone. Mortar is particularly susceptible.	Masonry spalling, loss of mortar
Weathering near to the portal	Seasonal and nightly temperature changes near to the tunnel portals causes expansion and contraction of brick and mortar at different rates. This damages the integrity of the masonry. Freeze thaw weathering also attacks the masonry mechanically.	Masonry spalling, mortar damage, cracking of masonry
Vibrations from trains	The regular passage of trains causes vibrations and cyclic loading (Degrande et al., 2006)(Degrande et al., 2006). This causes mechanical damage to the masonry.	Cracking of masonry
Overloading	Above ground construction and changes in the water table adjust the tunnel overburden pressure.	Masonry spalling, masonry block failure, lining deformation.
Changes in loading state	Voids forming behind the lining and changes in the lining profile lead to changes in the loading state, moving the thrust line of the arch.	Lining deformation, formation of lining bulges

Table 2.2: Common construction flaws in 19tth century railway tunnels that accelerate lining deterioration.

Construction flaw	Cause	Long term Impact
Unfilled shafts	Some construction shafts are left open for ventilation, but others are infilled after construction. However, poorly infilled shafts provide less protection from temperature changes and water ingress. Furthermore, if the shaft location is unknown and it's lining is poorly maintained, the shaft can collapse, leading to a change in loading on the tunnel's masonry arch.	Additional water ingress, freeze-thaw weathering and cyclic thermal weathering, changes in loading state.
Lack of mortar	In some locations masons may improperly apply mortar between masonry courses to save money or time	Masonry block detachment, decreased resistance to changes in loading state.
Poor connection between construction sections	Settlement between construction stages leads to discontinuities at masonry joints between tunnel sections. Masonry bonding was often not continued through adjacent sections leading to a reduction in shear resistance.	Greater susceptibility to cracking near to the section joint.
Poorly compacted fill behind lining	It was challenging to properly infill the area behind a constructed masonry arch due to varying amounts of overbreak. Varying compaction of the fill leads to an uneven amount of load transfer from the ground to the lining.	Uneven loading exacerbates later changes to the loading state.
Timber left behind the lining	Temporary support timbers may become stuck behind the lining and so cannot be slid forward to the next section. These are left in place after construction.	The timber will rot, leaving a void behind the lining. This changes the loading state.

2.3. Tunnel condition assessments

It is vital that any damage to a tunnel's lining is assessed to determine what mitigation work is required to maintain the tunnel's serviceability and ensure that it is safe to use. With most deterioration developing progressively over many years, regular tunnel condition assessments should be conducted to identify the location and severity of any damages and track the overall health of each structure (Network Rail, 2016). Changes to the lining condition must be correctly documented to enable an effective analysis of a tunnels

condition. If damages are incorrectly documented or unnoticed then deterioration may worsen, and later remediation work may need to be more extensive and costly. Ultimately, collapse and loss of life may occur if deterioration continues uninterrupted. Uncertainty in the condition can also result in overly conservative maintenance strategies, increasing cost and negatively impacting the long-term outlook of the railway network.

Within the UK, most railway tunnels are owned by the British railway asset manager, Network Rail. While there are no global masonry tunnel inspection standards, Network Rail's methods (Network Rail, 2023) are typical of those provided by other asset managers globally such as on the Paris metro (Llanca et al., 2017), by SNCF in France (Thakker et al., 2015) in Japan (Asakura and Kojima, 2003), in Sweden (Sjölander et al., 2023) and in Shanghai (Li et al., 2017). Tunnels in the UK receive yearly detailed structural assessments in addition to more regular visual assessments. Detailed assessments are intended to build a complete picture of a tunnels condition and inform future maintenance strategies, while visual assessments are intended to monitor the progression of defects and identify any immediate risks to a tunnel's serviceability. Visual assessments do not involve calculations and consist only of a short inspection.

Detailed condition assessments involve an onsite inspection followed by a structural assessment and the use of Network Rail's Tunnel Condition Marking Index (TCMI). The TCMI outputs a single value to quantify the condition of a tunnel. Assessors follow the guidelines provided within NR/L3/CIV/006 to record any disorders identified on site. The severity of each disorder is standardised and input into a spreadsheet to calculate the overall TCMI score for a tunnel. This score can be used to inform asset

management strategies, for example giving an indication of the urgency of repair work. The assessment process is visualised in Figure 2.3.

Despite the standardised procedure, detailed tunnel condition assessments are 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 unnecessary remediation works being conducted, or deterioration going unnoticed and requiring more challenging repairs in the future. To reduce the cost and environmental impact of additional work, it is important to improve the consistency of results. The unreliability of human assessors conducting visual inspections is a known problem across condition assessment disciplines. The issue has been noted by Laefer et al. (2010) for building crack detection and by Phares et al. (2004) for highway bridge inspection.

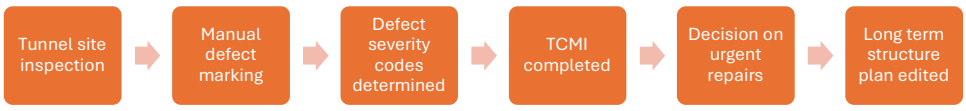


Figure 2.3: Network Rail's high-level workflow for a detailed tunnel condition assessment.

2.3.1. Tunnel Inspection

The tunnel inspection procedure consists of a team of inspectors traversing the tunnel and visually observing defects using torches. Voids behind the tunnel lining may be identified by tapping the lining with a listening tool, hearing for changes in the sound produced by the lining. A tunnel inspection must take place during a line closure, so is typically conducted at night to avoid costly operational disruption. A scaffold often needs to be erected to enable a more detailed visual and tactile inspection of the tunnel lining.

Photographs, written notes and diagrams of defects are made by the inspectors, but time limitations prevent these from being exhaustive. As a result, TLS (Terrestrial Laser Scanning) surveys are increasingly being undertaken during an inspection to enable a 3D point cloud of the tunnel to be generated. This enables many defects to be identified and analysed from an office later. Figure 2.4 shows a TLS survey being undertaken during a routine inspection of a masonry lined railway tunnel in the UK.

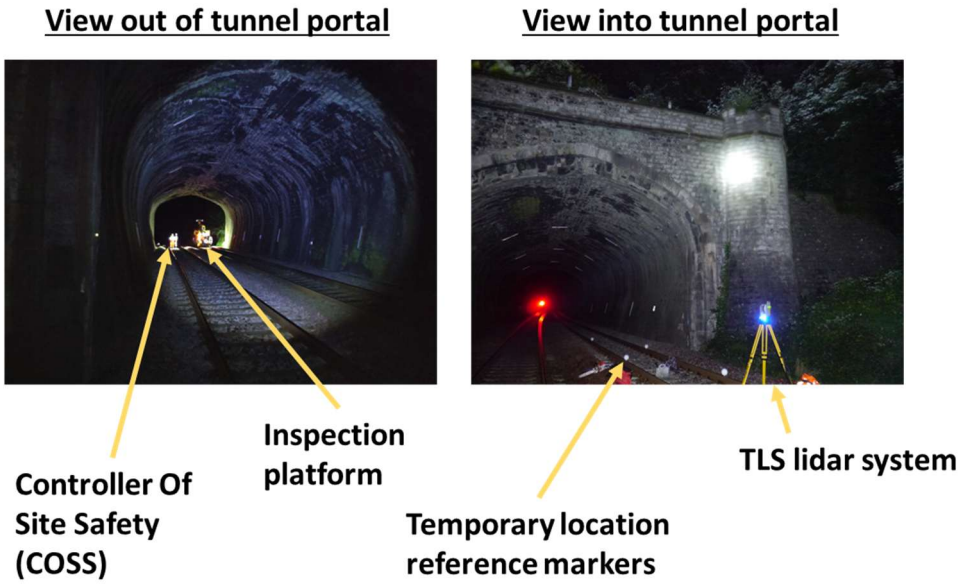


Figure 2.4: Photographs taken during a routine railway tunnel inspection in the UK, taken at night during a line closure.

2.3.2. Point cloud surveys

3D point clouds document the locations of georeferenced external observable points on a structure. They are typically obtained using phase comparison laser scanners. There are also time-of-flight based scanners that are commonly used in aerial surveys. Phase comparison is typically used for structural inspections as it can capture points with a higher frequency (though over a shorter range) than time-of-flight scanners (Suchocki, 2020).

Phase comparison works by sending out a continuous laser beam that is modulated at a constant frequency. The beam reflects off solid objects back to the scanner. The distance to the object that the laser hits can be calculated from the phase shift of the beam after it returns to the scanner. Accuracy decreases with increasing distance from the scanner. A typical scanner used for tunnel inspection, the Faro Focus series, (FARO, 2022) has a 3D position accuracy of 2mm at 10m distance and 3.5mm at 25m distance.

2.3.3. Defect identification and quantification

After the inspection, it is necessary to markup drawings of the tunnel with the exact location and nature of tunnel disorders. In accordance with NR/L3/CIV/006 (Network Rail, 2016)(Network Rail, 2016), there are 7 notable types of masonry lining defects as follows:

1. Spalling - This is when parts of the masonry face have broken off
2. Open joints or perished mortar - This is where the mortar no longer adheres properly to the masonry blocks
3. Water ingress - The masonry is damp or there is visible water flowing
4. Hollow sounding areas - There is likely an open void behind the lining
5. Bulges or lining deformation, distortion or flattening - The tunnel cross section has changed from as-built
6. Loose or missing masonry/ block loss - Blocks have become loose or fallen out of the lining
7. Cracks and fractures - There is a notable gap running around the lining

Each of these also have associated defect severity codes noting the extent of the defect, outlined in Table 2.3.

Table 2.3: Defects classified within the TCMI defined by Network Rail for detailed tunnel condition assessments in the UK.

Defect	Severity calculation	Potential causes	Remediation techniques
Spalling	Depth of spalling	long term damp, freeze-thaw weathering, chemical attack, lining overloading	Block replacement or surface patch repair
Open joints	Depth of lost mortar	long term damp, freeze-thaw weathering, lining deformation opening gaps between blocks	mortar repointing or block and mortar replacement
water ingress	Flow rate of water ingress	deterioration of masonry behind lining, change in ground conditions	lining replacement, grouting, hydrophilic resin
hollow sounding areas	Amount of lining distortion	masonry ring separation, ground movement behind lining causing a void	block pinning, further investigation using ground penetrating radar or ultrasound
lining deformation	Existing or new	ground movement or deterioration of the tunnel lining	Installation of structural health monitoring to measure changes in deformations, tunnel lining replacement, block pinning, rock bolts
missing masonry	Loose or missing	mortar loss, lining deformation causing gaps between blocks	lining replacement, shotcrete
cracking	Crack orientation and width	lining deformation, ground movements, water damage	infilling

The highest severity defects require immediate remediation, while the lowest severity ones will usually be left and monitored in subsequent assessments. In the case of spalling, the depth of spalling determines it's severity. This calculation is visualised in Figure 2.5 and forms one of the most time-consuming parts of a typical tunnel inspection given the large areas of low severity spalling typically present on Victorian era tunnels. The defined severity categories are outlined in Table 2.4.

The Network Rail standards assume that each masonry block can lose up to half of it's effective width before failure, so the most critical S1 damage threshold is set as 50% of a typical block's width. The impact of spalling damage on a tunnel lining's stability is therefore relative to the size of each block. Stone lined tunnels typically use larger masonry blocks than the bricks used in brick lined tunnels, so the spalling depth thresholds are set higher for each severity level.

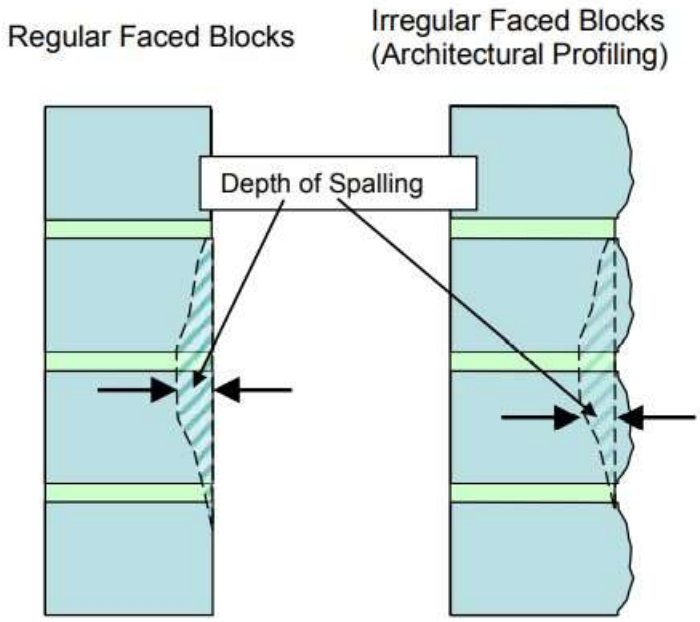


Figure 2.5: Visualisation of masonry block spalling depth calculation from Network Rail tunnel assessment standards (Network Rail, 2016)(Network Rail, 2016)

Table 2.4: Spalling severity categories as defined by NR/CIV/L3/006/4C (Network Rail, 2016)

Defect code	Depth of Spalling	
	Brickwork	**Stonework/Blockwork**
S1	> 50mm	> 100mm
S2	> 20mm and ≤ 50mm	> 40mm and ≤ 100mm
S3	≤ 20mm	≤ 40mm

Tracking the overall amount and severity of spalling is key to understanding the rate of deterioration of the masonry blocks. However, less severe spalling is typically not an immediate structural issue, so inspectors generally focus their attention on cracking and lining deformations. Deep spalling can be difficult to differentiate from lining bulges and can also obscure other damages (Chiu et al., 2015). Spalling severity is therefore often incorrectly or over simplistically mapped and simple, preventative masonry maintenance may be skipped. An example of a masonry spalling severity map labelled onto an image of a small area of tunnel lining is shown in Figure 2.6 and demonstrates the extent of work required given the number of masonry blocks within a tunnel. There is a clear need to reduce the labour required to collect accurate spalling data.

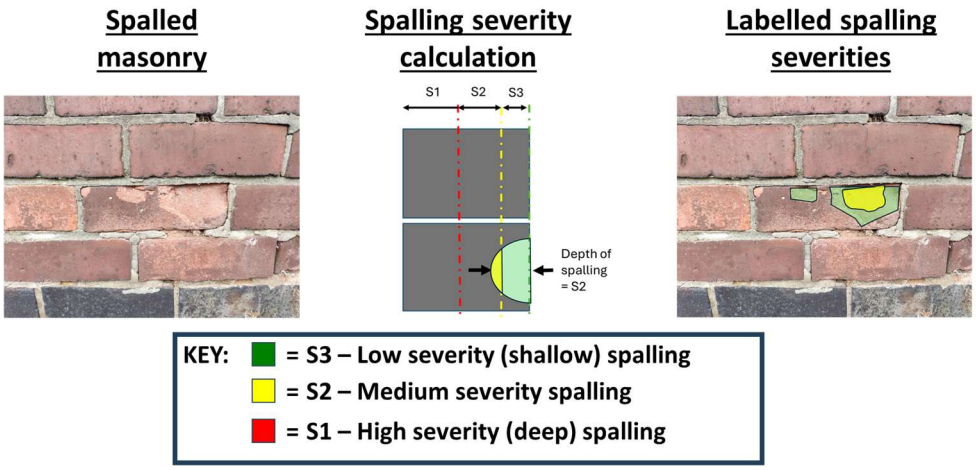


Figure 2.6: Example area of masonry showing how the spalling severity is defined by depth. a) Photograph of spalled masonry. b) Simplified diagram of a masonry lining cross section showing an area of spalling and the depth thresholds used to categorise spalling severity. c) Created spalling severity map overlaid on photograph of tunnel masonry,

Lining deformation is usually the highest risk defect, as it shows that the thrust line of the tunnel has moved and the lining is closer to forming a collapse mechanism. While often difficult to identify on site, deformation can be identified by analysing the lining profile on a 3D point cloud obtained by lidar. However, it is difficult to differentiate lining deformation from deep masonry spalling.

Spalling localisation and categorisation can also be conducted by analysing a 3D point cloud. A defect depth value is assigned to each point and thresholding is used to classify the areas of spalling. The defect depth is calculated as the offset of the point cloud from a best fit mesh which represents the undamaged intrados surface. The profile of the undamaged intrados is then used to identify lining profile abnormalities and deformations. To differentiate between areas of spalling/masonry loss and tunnel lining deformation, the mesh should follow the deformed shape of the tunnel and not the as-built shape. However, this 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. Secondly, mortar and crack locations need to be manually masked out to only consider block face areas and isolate brick spalling from mortar loss or masonry cracking. Creation of the mesh is a manual process, based on trial and error. Various smoothing and fitting methods are employed to capture the shape of the tunnel, while avoiding areas of masonry damage. As shown in Figure 2.7, it is challenging to accurately fit the mesh to the undamaged profile.

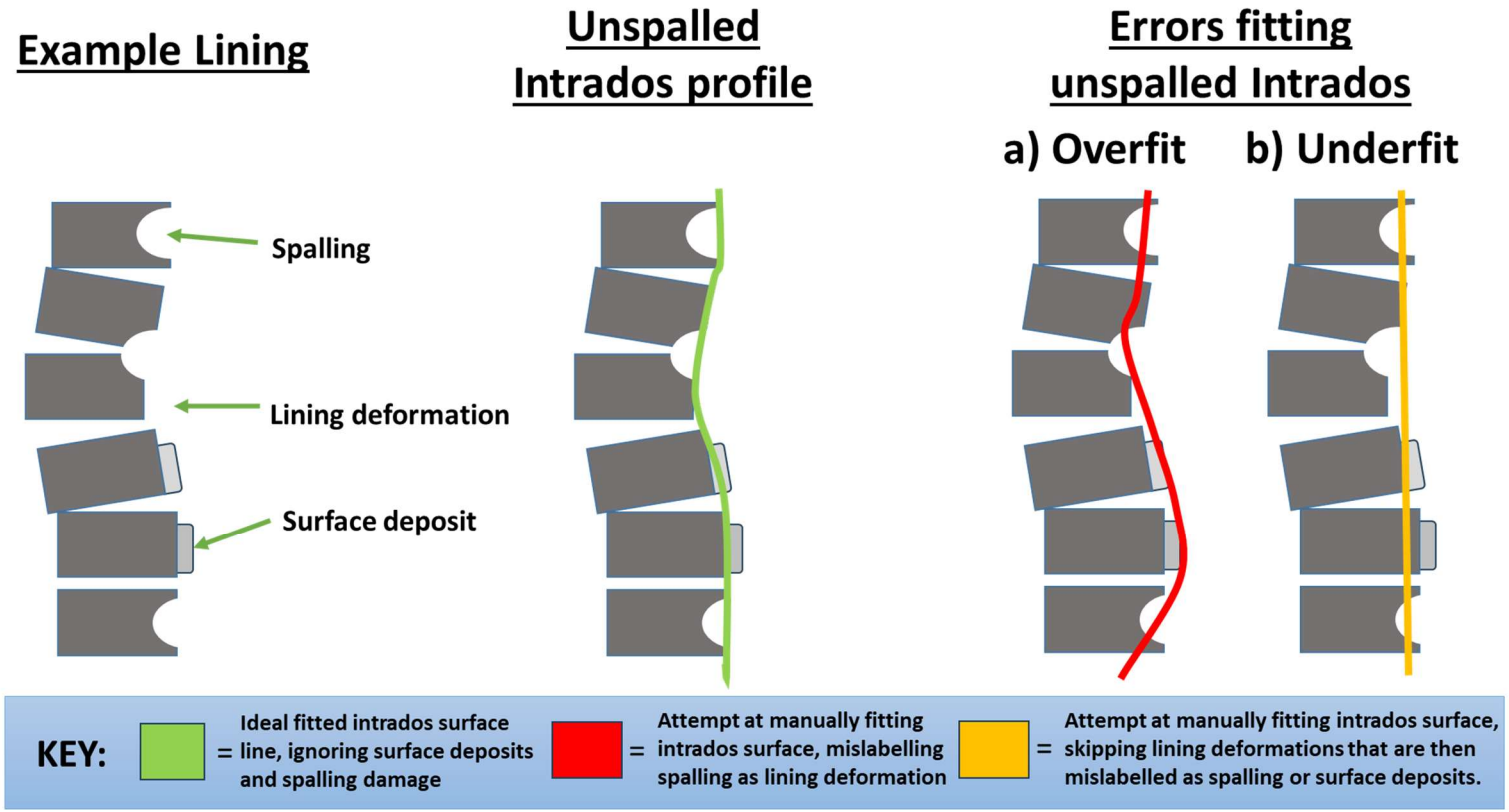


Figure 2.7: Visualisation of the challenges to separating lining deformation from spalling on a tunnel lining point cloud. The figure shows a simplified example of a section of deformed tunnel lining with spalling and surface deposits present on some of the blocks.

Missing masonry and large cracks may also be identified from the point cloud. Mortar damage and hairline cracks are too small to be identified within the accuracy of a lidar survey. As a result, localisation of these is dependent on the inspectors' notes. Calcite staining indicating water damage may be identified from the RGB data of a lidar survey, although the appearance of this is variable and dependant on the masonry type and the surrounding rock formation.

After the assessment, the assessors will recommend remedial works to be undertaken on the structure together with a level of urgency. This is aided by the tunnel condition score obtained by entering each aspect of the structural condition into the TCMI. This is again based upon the assessors engineering judgement and, as noted in Forster and Douglas (2010), is a subjective process. Network rail will input the TCMI score into a database and use it to identify which structures need repair work prioritising and which will require larger scale renewal or replacement works. Such databases are also useful for higher level data analysis such as that conducted by Sandrone and Labiouse (2011) on Swiss road tunnels to identify the most common drivers of deterioration. These studies can be used to guide the direction and strategies of the maintenance funding organisation. Asset managers may also use the TCMI as input to decision support tools such as PADTUN (Thakker et al., 2015) to guide maintenance scheduling. Developed for the French railway operator SNCF, PADTUN is an intelligent system designed to pool expert knowledge into a single automated database that can identify underlying tunnel pathologies from identified defects.

2.4. Key Challenges

Railway infrastructure managers globally are facing conflicting demands of increasing railway traffic and decreasing costs while ensuring the safety of their customers and employees. Given that 19th century masonry lined tunnels make up a considerable proportion of the UK's railway infrastructure assets by milage and are operational on most railway networks globally, it is essential that masonry tunnel condition assessment is made more efficient and effective. Reducing the level of subjectivity inherent in current inspection and assessment practise will improve assessment repeatability and reduce uncertainty, enabling more targeted maintenance interventions. This will improve tunnel safety and reduce costs, so that infrastructure budget can be allocated to the network expansion and improvements that are required to encourage a modal shift of passengers and freight back to rail. Three key challenges have been identified that are limiting the performance of condition assessments:

- Limited time on site due to labour costs and the need to work outside of railway operation times curtails the scope of data collected on site.
- Spalling severity identification takes up a substantial proportion of an inspector's time, limiting the capacity to inspect more critical or unusual damages such as lining deformation or cracking which require more expert input.
- Work to digitalise aspects of an inspection, such as spalling and deformation identification from lidar surveys, has been limited by the lack of digital tools and workflows to analyse the data in a timely and repeatable way.

As a result, there is a need for a consistent workflow to more efficiently analyse a masonry tunnels condition. Lidar surveys provide accurate documentation of a tunnel's lining and are an ideal data source for an automated analysis workflow. There is demand for a method for accurate classification of masonry spalling severity locations and lining deformations directly from the generated 3D point cloud. This would provide verification of the defects identified in the on-site inspection and generate a more comprehensive description of lining damages. Automation of this workflow would reduce costs and subjectivity.

3. Automated condition assessment: An overview

3.1. Digital Tunnel inspection techniques

The typical tunnel condition inspection process employed by industry is highly manual and the working conditions inside railway tunnels are often poor. Tunnels are dark, humid and cold. Railway track can be difficult to walk along, with hazards from tripping and falling objects when tunnel linings are disturbed during tactile inspection. As a result, there is both a cost and Health and Safety incentive to minimise the labour time required within the tunnel. Differences between the format and brevity of collected data and site notes also often makes it difficult to track changes in the tunnel between inspections. In addition, new methods of tunnel inspection may enable better detection and more accurate description of disorders.

Throughout the tunnel condition monitoring and maintenance process there are opportunities for the application of new technologies and methods. Despite this, aside from the introduction of photographic documentation and, more recently, lidar surveys, the assessment process adopted by industry has modernised little compared to those within other engineering disciplines. Driven by recent innovations in AI (Artificial Intelligence) and robotics, multiple studies have been undertaken to improve the condition assessment process of concrete tunnel linings (Sjölander et al., 2023). Few studies have investigated masonry lined tunnels, although there has been research targeting damage detection and monitoring in masonry structures more generally (Soleymani et al., 2023). There are, however, many more technologies that have already proven effective at automating or otherwise improving the assessment of other civil infrastructures (Agnisarman et al.,

2019b). Bridge condition assessment has received especially strong research interest (Ahmed et al., 2020) and many techniques developed for monitoring the underside of a bridge may be applicable to monitor a tunnel's lining condition.

This chapter first outlines methods for collecting digital tunnel condition data and those with potential for use in routine condition inspections. The Chapter provides an overview of past work to either automate inspection tasks or provide data in a format that can streamline subsequent analysis tasks. Automated defect detection and analysis methods are then discussed in detail and the most promising methods for masonry tunnel analysis are evaluated.

3.1.1. Photographic data collection methods

Beyond manual notetaking, photography is the most common method of recording the state of a tunnel lining. Tunnel inspectors take photographs of the tunnel lining to document the nature of each recorded damage location. Usually, these photographs are taken ad hoc at damage locations and linked to site notes to provide a visual description of damages, although occasionally a comprehensive set of photographs of the tunnel will be taken to enable further identification of defects post inspection from an office setting. This would provide a complete reference of the tunnel lining's state at the time of the inspection. However, without a costly specialized camera trolley, it is difficult to ensure the field of view geometry and lighting is standardized making it difficult to stitch the photos together into a single lining panorama without using feature matching algorithms (Attard et al., 2018). Despite extensive research by (Chaiyasarn, 2014) to develop improved algorithms for tunnel lining image stitching, the quality of a lining panorama

created from individual DSLR images still lags behind using dedicated panoramic camera hardware. Furthermore, it is challenging to manually collect enough photographs to cover a longer tunnel during an inspection.

3D point clouds are increasingly being created during inspections to document a tunnel's lining. A 3D point cloud may be generated using lidar or photogrammetry and consists of a set of 3D co-ordinates representing external points on the target structure. Generating a point cloud provides multiple benefits for tunnel condition inspections (Dekker et al., 2023):

1. If the 3D accuracy of the point cloud is high enough, then accurate measurements of tunnel parameters and defect locations can be made digitally.

2. The point cloud provides a useful 3D visualisation of the tunnel lining. This helps assessors to visually identify abnormalities in the lining profile and makes it easier to spatially relate damages with drivers of deterioration.

3. Subsequent point cloud surveys can be taken of a structure to investigate the progression of deterioration. If both surveys are aligned, then they can be compared and any changes to the geometry of the structure can be measured to track the rate of deterioration.

4. Photographic colour data can be projected onto the point cloud. This creates a visually authentic model that can be used to better communicate and visualise details within the tunnel.

5. The comprehensive and standardised nature of 3D point cloud models form the ideal input to computer vision algorithms for automating the identification of tunnel damages. A point cloud can also be used to inform the creation of a Building Information Model's (BIM) geometry, creating a digital twin of the structure.

Using photogrammetry, photographic data can be used to generate 3D point clouds of a structure. Photogrammetry is the practise of extracting geometry measurements from photographs and has existed in different forms since the invention of photography in the mid-19th century (Saif, 2022). Structure from Motion photogrammetry (SfM) involves taking overlapping photos from different locations and angles (Granshaw, 2018). Feature tracking algorithms then match locations between photos in order to calculate the relative perspective of each image. This enables matched pixels in each image to be projected into 3D space, generating a 3D point cloud of the target object (Westoby et al., 2012).

Before the advent of modern SIFT (Scale-Invariant Feature Transform) algorithms as proposed by (Lowe, 2004), photogrammetry required the knowledge of the location and orientation of the camera for each photo. SIFT was the first algorithm to robustly identify features at different scales, orientations and levels of illumination, causing SfM to gain popularity for its increased ease of use. Any device with a camera can now be used to record data and there are many accessible software packages available for processing the images (3DSourced, 2024). However, despite research by (Holynski et al., 2020), SfM suffers from drift when used over long repetitive or featureless scenes such as tunnels. This is where the global accuracy of the output geometry decreases for data captured further from a fixed reference point, although the produced model retains its local accuracy relative to neighbouring locations. Drift occurs because small errors in the photograph registration accumulate over the length of the tunnel and, even if the start and end are georeferenced, there are few features out of plane of the tunnel lining to use as radial coordinate reference points.

3.1.2. laser based data collection methods

The 3D accuracy of lidar has led to its increasing popularity for generating 3D point clouds of civil infrastructure. There has been extensive research on using lidar throughout the broader field of structural condition assessment (Kaartinen et al., 2022). The most used type of lidar for tunnel inspections is phase-based TLS (Terrestrial Laser Scanning). This involves setting up a lidar scanner on a tripod at different locations inside the tunnel. Multiple scans will be taken, then manually registered together by point matching. Typically, 5m intervals are used to generate a registered scan that can achieve a maximum 5mm point spacing with a standard TLS system such as the Faro Focus (FARO, 2022).

In addition to collecting 3D location data, lidar systems also record the signal intensity of each measurement. This is the proportion of the power of the emitted signal that is returned to the scanner. While this value is not typically used for data analysis, (Antón et al., 2022) have shown the intensity value to be effective in identifying material types and highlighting surface defects. In addition, (Höfle and Pfeifer, 2007) demonstrated that with additional processing, the reflectance of the material of each recorded point can be inferred from the intensity. This can help determine the material's condition, by indicating its surface roughness. TLS systems may also be equipped with other sensors such as RGB (3 channel Red, Green and Blue colour) or Infrared (IR) cameras. This sensor data is projected onto the lidar generated point cloud.

Despite improvements in SfM photogrammetry algorithms, the predictable performance of TLS generated point cloud data has led to TLS being the most commonly used method of obtaining 3D tunnel data (Attard et al., 2018). Even over small areas, (E. Valero et al., 2018; Valero et al., 2016)

demonstrated that current industry standard Terrestrial Laser scanners create 3D point clouds with higher levels of 3D accuracy and precision than photogrammetry taken with a high quality DSLR. (Attard et al., 2018) reviewed the use of photogrammetric techniques for tunnel inspection and concluded that a principal reason for the lower performance and resulting lack of adoption of camera based methods compared to laser ones is due to the low and variable light levels within a tunnel in addition to the often relatively featureless lining surfaces rendering automatic patching together of images difficult.

The quality of DSLR photogrammetry varies substantially, depending on the texture of the surface. Nevertheless, colour information obtained from the inbuilt camera in most TLS systems is typically much lower quality than that obtained by consumer DSLRs and the differing location of the RGB and lidar modules on a TLS system can lead to colour projection errors on the generated point cloud. Mapping of colour data coherently to the generated point cloud is inherent when using photogrammetric methods. (Zhang and Lin, 2016) reviewed the methods of fusing photographic and lidar data in order to provide more comprehensive documentation of tunnel surfaces. In practise, the benefits of data fusion methods often do not justify the additional cost of collecting multiple datasets.

Mobile Laser scanning (MLS) is popular for conducting rapid structural inspections (Babic et al., 2012). MLS creates a 3D point cloud model while the laser scanner is in motion and is used to map large areas, by mounting laser scanners onto cars or Unmanned Aerial Vehicles (UAVs). It is commonly used in object detection workflows within autonomous vehicles. Each point is Geo-located relative to the vehicle's location, which is determined from GPS data that has its accuracy enhanced by correlation

with outputs from an inertial measurement unit (IMU). An issue with this method is that it requires a GPS signal or other location tracking system to function. While this is difficult to achieve in a tunnel, there are alternative methods for geolocating an MLS system by mounting it to a train. Knowledge of the railway track centreline can be combined with speed information from a train's odometer (Dong et al., 2022). The (Fugro, 2022) system, for example, mounts photography and lidar equipment on to in service trains to gather data on lineside structures without having to close the railway or send any personnel. It has been applied in industry to conduct gauging measurements and identify railhead and OLE defects.

SLAM (simultaneous location and mapping) enables mobile laser scanning to take place underground when global geolocation is not available. A SLAM system determines its location by applying algorithms to cross reference newly acquired point cloud data with points acquired when the scanner was at previous locations. An outline of the field can be found in (Cadena et al., 2016; Riisgaard and Blas, 2005). SLAM systems have gained substantial popularity for their ease of use; Surveys only require someone to carry a handheld scanner (for example, the GeoSLAM ZEB Horizon RT (GeoSLAM, 2022)). Unfortunately, similar to photogrammetric methods, the accuracy of SLAM systems decreases as they are moved away from fixed reference points and the system’s position calculation drifts from the true location. The loss of accuracy is particularly pronounced in a railway tunnel environment where there is a relatively featureless interior and errors compound over the long distances between external reference points. In a study by (Trafikverket, 2017), a 55kmph drive through a 1.7km long concrete lined tunnel produced a maximum SLAM drift of 0.4m for a vehicle mounted system. (Liu et al., 2025) showed even greater drift with a handheld system, developing 2m of

drift over a 333m long tunnel section. While SLAM system accuracy may be improved through recalibration against fixed targets (Keitaanniemi et al., 2023), the added time and cost of conducting an additional TLS survey for calibration largely negates the time saving benefits of SLAM. There has been extensive research to reduce the effect of drift by, for example, (Tang et al., 2018; Tian and Ma, 2018) ,but the produced scan accuracies are not yet sufficient for SLAM to be applied to structural condition assessments.

The point density and precision of SLAM and MLS systems are also limited by the speed and smoothness of motion of the scanner and ultimately TLS may be required for smaller defects such as cracks or mortar loss to be effectively identified and categorised from a generated point cloud.

3.1.3. Specialised data collection

A substantial amount of information about a tunnels condition can be inferred from the condition of the lining surface. In order to investigate the causes of lining damages, it is helpful to understand what is happening inside the lining or behind it. Voids behind the lining caused by ground movements, poorly compacted fill or unblocked shafts can change the lining loading state and cause collapse. However, these can only be inferred from a 3D point cloud once lining deformations have already started to occur. The traditional industry method of identifying voids involves manually tapping the tunnel lining and listening for changes in the sound produced that may signify lining deterioration, a change in fill density, or a void. However, this method is strongly subjective; identifying the exact nature of any abnormalities heard is challenging. Further destructive investigation into the lining, such as by coring, risks damaging the structure and is unsustainable to conduct in repeat inspections. There are multiple non-destructive methods designed to

identify the condition of areas underneath the tunnel lining (Richards, 1998). Methods have also been developed for other types of masonry structures (Hussain and Akhtar, 2017; Schuller, 2003).

Ground Penetrating Radar (GPR) has been used for imaging the subsurface in many engineering disciplines (Solla et al., 2021). Within new build tunnels, it has been applied to detect voids in grouting (Kravitz et al., 2019). GPR uses microwave band radar pulses to detect changes in material properties from the reflected radar signal. Performance is dependent on the selected radio frequency, which is chosen based on a compromise between signal resolution and penetration depth. Changes in material properties can only be identified from the output radiogram by a trained expert and there are often multiple explanations for a given radiogram. While AI-based approaches to automate radiogram interpretation show promise, the field is still in its infancy (Küçükdemirci and Sarris, 2022). Despite these challenges, (McDonald et al., 2023, 2022) show that the T-Vision helical GPR system can effectively image the subsurface inside and beneath a tunnel's masonry lining.

Transient Electromagnetic Radar (TER) (Geng et al., 2022) has been proposed as an alternative to GPR to identify voids behind concrete tunnel linings. TER emits a pulsed magnetic field and records the returned magnetic field created by eddy currents induced in the target object. The size of the eddy currents is proportional to the resistivity of the object's material. TER was shown to more clearly differentiate the different layers in the lining and boundaries of voids over 10cm diameter than GPR. In addition, it was less impacted by the presence of steel rebars in the concrete lining.

Ultrasound (Carlson and Jansson, 2021), can also be used to detect material density changes. However, ultrasound systems must make contact with the

tunnel's lining making them more cumbersome to use around the crown. Their output also requires expert interpretation.

Passive infrared thermography (IRT) uses the tunnel lining surface temperature to detect defects. Thermal cameras were first used by (Konishi et al., 2016) to identify voiding in concrete tunnel linings as a precursor to spalling. Voided areas had a lower temperature than intact concrete due to the lower temperature conductivity of air. The study found that the tunnel lining needs to be at least 0.35°C warmer than the air temperature inside the tunnel for there to be a great enough temperature gradient to make the surface temperature variations visible. A similar method has been applied by (Behbahani et al., 2024) to identify debonded tiles in tiled underground passageways. Thermography can also be used to identify water ingress in concrete lined metro tunnels. (Yu et al., 2018) fused lidar intensity with thermal camera data to identify wet areas on a tunnel's lining. Wet areas of concrete are more reflective, so the intensity of the returned lidar pulse is different to that in dry areas. Due to evaporation in the wet areas removing heat from the lining, the wet areas will show a lower temperature than dry areas in the thermal images.

A final emerging method that can be used for identifying hidden unblocked shafts is Muon tomography (Thompson et al., 2020). Muon tomography records the number of muons, a type of cosmic ray, incident onto a recorder inside the tunnel. Muons vary in energy, with the lower energy ones more likely to be absorbed in the tunnel overburden. The rate of muon incidence is therefore dependant on the tunnel depth and overburden density. It can therefore be used for the estimation of tunnel overburden density when the depth is known (Zhang et al., 2025). For a reliable model to be generated, the low frequency of muon incidence results in a considerable data collection

duration. With a muon incidence rate of around 70 per minute at around 20m depth, (Thompson et al., 2020) for example, required 100 hours of data collection to reliably identify the location of a void behind the lining of a 770m long tunnel. Furthermore, it is only able to record changes in density along a tunnel's length and cannot segment the circumferential location of abnormalities.

While these non-destructive methods are occasionally used for in depth structural investigations, they require specialist equipment and operators, so (Richards, 1998) note that their use is not routine. A final category of lining testing are minimally destructive methods. While these methods are invasive, if they are correctly applied the impact on the tunnel's condition should be negligible.

(Llanca et al., 2013) expanded the use of geoendoscopy to masonry tunnel linings. They placed an endoscopic camera inside 6mm diameter boreholes to image the subsurface inside and behind the masonry lining. Using simple image thresholding, they were able to identify discontinuities within the lining and surrounding ground.

An impulse response test can also be used to examine the integrity of the ground behind the tunnel lining. (Davis et al., 2005) first used the method to identify grouting voids at the concrete tunnel lining / ground interface. Conducting an impulse response test involves applying a controlled impact to the tunnel lining, usually with a 1kg sledgehammer. A velocity transducer is then used to analyse the vibration response of the lining to the impact, which will be affected by material properties and reflections of the stress wave at material boundaries. Frequencies below 50 Hz can be used to calculate the structural element's dynamic stiffness, while frequencies from 100 Hz to 800 Hz enable changes in density and thickness to be identified.

The Impulse response test is effectively a more sensitive, scientific and standardised version of the traditional tapping and listening method that tunnel inspectors routinely apply to identify voids behind the tunnel lining. While Impulse response is proven effective at identifying voids behind tunnel linings (Cao et al., 2019), using it to detect deterioration of the lining would first require accurate characterisation of the lining material which would be difficult to achieve given the non-homogeneous nature of most historic masonry linings. (Llanca et al., 2017) showed how GPR, impulse response and endoscopic inspection could be combined into a comprehensive condition assessment workflow for masonry lined Paris metro tunnels. While their method was shown to effectively identify lining discontinuities and characterise the surrounding soil, it does not consider either localised surface damage such as spalling or abnormalities in the lining profile.

In order to track changes in a structure's condition, SHM (Structural Health Monitoring) methods can be used. SHM typically involves mounting various sensors to a structure and tracking changes in their readings either through periodic inspection or a remote data connection. Abnormal readings may signify deterioration of the structure. While there has been extensive research to use the dynamic response of bridges (Wan et al., 2023) to predict their serviceability state, similar applications within tunnels have been limited (Chen et al., 2022; Nellen et al., 2000). However, other SHM methods are often integrated into new build tunnels, with fibreoptic monitoring cast inside concrete linings for measuring lining or reinforcement strains (Chen et al., 2025). Vibrating wire sensors may also be used to measure lining cracking or displacement. For existing tunnels, simple crack monitoring gauges such as Tell-Tales ("Plus Tell-Tale - Avongard," 2025) may be placed across tunnel discontinuities such as cracks or open joints and used to look for lining

deformations by comparing the recorded relative displacements between subsequent visual inspections.

An overview of alternative tunnel defect data collection techniques is provided by (McCormick, 2010) and, more recently, (Strauss et al., 2020). (Haack et al., 1995) provides an older and broader overview. Research activity has been extensive, but the expense and lack of knowledge around using more specialist methods means that it is unlikely that they will be regularly used in older masonry lined tunnels outside of targeted structural studies unless their use can be more standardised or automated.

3.1.4. Robotic tunnel inspection

Considerable research has been undertaken across engineering disciplines to reduce labour costs and the need for substantial domain knowledge by inspection staff by developing robotic methods of infrastructure inspection (Lattanzi et al., 2017). (Montero et al., 2015) outlines various robotic machines that have been created specifically for tunnel inspection and maintenance, mostly for concrete-lined road tunnels. Robotic methods reduce the need for inspectors to work at height when observing a tunnel's crown or enter tight spaces, contributing to improved health and safety outcomes. Most proposed robotic inspection systems involve mounting various sensors, such as ultrasound or GPR onto robotic arms to closer inspect concrete tunnel linings. Such systems are expensive to design and maintain. The issue of transporting them to inspection sites means that they can only be justified in areas with a high density of tunnels, for example in underground metro systems.

Robotic systems currently under development include the European commission funded ROBINSPECT (ROBotic System with Intelligent Vision

and Control for Tunnel Structural INSPECTion and Evaluation). Outlined in (Loupos et al., 2016), this ambitious project aimed to mount lidar, photogrammetry and ultrasonic sensors onto a semi-autonomous inspection vehicle. The robot would enable disorders to be automatically detected, closely inspected and analysed with limited human intervention, generating a 3D model of a tunnel with labelled defects. To date, only a prototype of the vehicle has been created along with an outline plan for automation of the system. More recently (Liu et al., 2023) proposed a rail mounted tunnel inspection robot that uses MLS for data collection. A speed planning and tracking algorithm was developed that paves the way for unmanned inspection vehicle control.

UAVs and wall climbing robots have been proposed for tunnel lining inspection (Montero et al., 2015). A key issue with using UAVs for tunnel inspection is the near-wall effect noted by (Robinson et al., 2016), which upsets the stability of flying vehicles in confined spaces. Despite this, (Vong et al., 2018) developed a UAV that performed satisfactorily in small culvert tunnels. This was achievable due to the decrease in size and weight of sensors in recent years, allowing the UAV to be smaller relative to the size of the tunnel. In this case, the UAV used measured only 375mm diameter, with a payload of 800g.

(Bendris and Becerra, 2022) equipped a UAV for railway tunnel inspection. Depth and tracking cameras and a distance sensor were used for mapping and navigation. A visual SLAM algorithm was used for positioning. Additional visual and thermal cameras were mounted for collecting inspection data. An algorithm was created to calculate the optimal flight path for conducting an inspection, thus enabling autonomous operation. Overall, due to the less constrained environment and better access to positioning data through GPS,

UAVs have been used more extensively for bridge inspections (Feroz and Dabous, 2021) than tunnel inspections.

Other robotic inspection vehicles include a wall climbing robot developed by (Garrido and Sattar, 2021) to detect rebar corrosion using GPR. This used magnetism to adhere to the rebars, allowing it to climb reinforced concrete structures. Wall climbing robots are less suited for masonry inspection, since they may cause additional tunnel lining deterioration and have difficulty adhering to damp and crumbling areas of tunnel lining.

At present, MLS, SLAM and photogrammetry are not commonly used in practical tunnel inspections due to their lower accuracy than TLS. The specialist operators and equipment required for GPR, Muon tomography and other specialist survey methods also result in their use not being routine. However, they are occasionally utilised on particularly challenging structures, when extra information is required to make an assessment decision. UAVs and robots are still in their early stages of application to infrastructure condition assessment, and the technology has not developed enough to be convenient for routine tunnel assessments. Another issue with large inspection robots is that the density of tunnels in the UK may not justify the logistical challenge of transporting them to each inspection site. An overview of data collection methods for tunnel inspection is outlined in Table 3.1.

Driven by the project-based nature of most condition assessment contracts, there are also multiple institutional reasons that slow the adoption of new methods in the construction sector. These include a low appetite for risk and limited innovation budgets (Elliott and Olbina, 2023). Changes to the established inspection procedure may also require cultural changes at the infrastructure provider. Staff would need to be trained to use any new technologies and there would need to be acceptance by the workforce of

changes in procedure for them to be effective in saving manpower costs. As a result, it is determined that there are more opportunities for improvements in the defect detection and severity analysis step of the condition assessment procedure than at the inspection stage which will be driven by the inspection equipment manufacturers.

Masonry-lined tunnel inspection is likely to be based around laser scanning backed up by local photographic documentation for the near future and improvements in the accuracy and convenience of TLS systems are likely to be incremental as commercial lidar stations become faster and more accurate. MLS and SLAM may eventually take over as the preferred methods once the algorithms have been refined to reduce drift and improve accuracy to an acceptable level.

3D point clouds generated by TLS typically require significant processing before being suitable for direct input into automated methods. Software such as Recap Pro (“ReCap Pro | Autodesk,” n.d.) provide a semi-automated human in the loop work flow for point cloud preparation, although it still takes around an hour to process a 20m long section of tunnel point cloud obtained at a 5mm point spacing with standard office equipment. Tasks include removing outliers and connecting up surveys scanned from different locations. Nevertheless, it would be beneficial to utilise point cloud data for automating condition assessments, since 3D data should ultimately be able to provide a more comprehensive analysis than 2D images. As Lidar surveys are typically already conducted during tunnel inspections, no new equipment or changes to the inspection procedure needs to be made

Table 3.1: Tunnel inspection methods.

Method	Description	Output	Advantages	Disadvantages
		Traditional Methods		
Visual inspection	Recording descriptions of visually identified damages in a notebook by hand	Set of handwritten site notes and diagrams	An experienced engineer can cover a variety of damage types and focus on the most critical locations	Highly subjective to engineer's judgement, Variable output between engineers, Damages may be missed, Notes may be difficult to understand
DSLR Photography	Photographs of individual defects	Colour photographs	Cheap, familiar equipment, Visual record of damages	Difficult to record location of photograph, Can be difficult to interpret out of context
Coring	Collecting a material sample of the lining and subsurface	Log of material layers beneath lining	Detailed record of material properties beneath lining surface	Damages the lining integrity, Only gathers information at location of core
Tapping	Tapping the tunnel lining, listening for changes in the sound produced	Handwritten descriptions of identified tunnel sounds and possible causes	Cheap method of indicating void locations behind the tunnel lining.	Requires an experienced listener, It is difficult to hear abnormalities and challenging to Identify the source of changes
		3D capture methods		
Terrestrial Laser Scanning (TLS)	Laser scan from fixed locations	3D point cloud	high accuracy, Complete 3D model	Requires supervision and multiple setup locations, only captures visible areas
Position based Mobile Laser Scanning (MLS)	Laser scan from a moving vehicle	3D point cloud	Survey conducted without tunnel closure	Positioning difficult with no GPS signal. Lower point density
Simultaneous Location And Mapping (SLAM)	Laser scan whilst in motion using algorithms for positioning	3D point cloud	Easy to use, lightweight	Drift reduces accuracy in tunnels, Lower point density
Structure from Motion (SfM) photogrammetry	Photos stitched together using algorithms	3D point cloud	High quality RGB data, Cheap, Easy to use	Lower 3D accuracy than laser scanning, Computationally expensive

Method	Description	Output	Advantages	Disadvantages
Specialist data collection methods				
Ground Penetrating Radar (GPR)	Reflected radar detects subsurface features	2D radargram into the tunnel lining	Detects area behind tunnel lining	Requires expert interpretation and analysis, Dependent on subsurface properties
Ultrasound	Records reflected high frequency sound waves	2D density map into the tunnel lining	Detects density changes and voids beneath lining	Requires surface contact, Maps require expert interpretation
Thermography	Infrared photographs of the tunnel lining	2D infrared heatmap of the tunnel lining surface	Temperature variations may indicate damp lining areas	Photograph locations need to be accurately recorded, Clarity will depend on lining surface characteristics
Muon tomography	Measures frequency of incident muons from cosmic rays	Estimate of overburden	Records data from above the tunnel - useful for detecting shafts	Data can be noisy, Requires large equipment, long recording times
Endoscopic inspection	Insert an endoscopic camera into a small diameter hole in the tunnel lining	Photographs of material inside and behind the tunnel lining	Can understand the masonry and backfill state cheaply and without causing damage, Boreholes may be reused	It is difficult to reliably assess the material properties from the photographs, Data only collected at borehole location
Impact- echo	Apply a controlled impact to the lining and measure the vibration response	Time–frequency spectra at set locations	Effective for triangulating void locations from only a few data points	Will not work well in heavily jointed materials or those with varying properties such as historic masonry
Emerging data collection methods				
Unmanned Aerial Vehicles (UAV)	Sensors attached to a drone	Typically only photographs, but other attachments are possible	Close up view of tunnel crown, Moves quickly	Limited sensor carrying capacity, Control is challenging in an enclosed environment
Robotic vehicles	Sensors attached to a robot	Any of the above + tactile inspection	Reduces manual labour, Potential for autonomous inspection	Large and expensive, Requires expert supervision and maintenance

3.2. Automated post-inspection tunnel analysis

Tunnel defects are identified manually on-site during tunnel inspections and in the office post-inspection through analysis of 2D and 3D sensor data. It is vital that defect identification is digitalised, as automated digital analysis workflows enable better standardisation and traceability of the reasoning behind maintenance recommendations (Weise et al., 2023). Current manual approaches are heavily reliant on the engineering judgment of individual assessors. Differences in the thoroughness of damage identification and opinions on the structural impact can compound into substantially different conclusions about a structure's prognosis (Forster and Douglas, 2010). Being able to automatically detect and assess defects from recorded data would enable rapid and comprehensive assessments and reduce the subjectivity of the classification process. Automated methods may also be easily adaptable to changes in assessment guidance brought by advances in our understanding of tunnel behaviour and the science of structural analysis. Generating consistent and reliable automated structural condition assessments would pave the way for more reliable predictive maintenance strategies (Lei et al., 2023; Tichý et al., 2021), reducing cost and improving safety across an asset manager's portfolio.

Computer vision offers the potential to automate the analysis of existing 2D and 3D models of a tunnel. Computer vision involves applying algorithms to automatically make conclusions about a scene directly from an image. In recent years, with the advent of deep convolutional neural networks (CNNs), the field has developed rapidly (Chai et al., 2021) and there is now potential for automated computer vision techniques to be more consistent in classifying defects than human assessors. This development has enabled many new applications and, as reviewed by (Spencer et al., 2019), computer

vision involving deep learning has been applied extensively for automated defect detection across a variety of infrastructure condition assessment tasks. Supervised machine learning methods using encoder-decoder based CNNs forms the most popular approach. The models must first be trained on relevant data to adapt them to the target problem. They can then apply their learnt knowledge to make conclusions about a new set of data. An introduction to computer vision methods using machine learning is supplied in more detail in Appendix A1.

There has been a particularly significant quantity of research into automating the process of crack identification and measurement (Hsieh and Tsai, 2020; Kheradmandi and Mehranfar, 2022; Munawar et al., 2021; Zhou et al., 2023), since it has applications across many engineering disciplines. A common defect, identifying and quantifying the size and direction of tunnel lining cracks is a critical part of determining a tunnel's structural condition. In general, cracks can be used as an early warning of structural deterioration as they may signify ground movements behind the tunnel lining. Once a crack has formed, it may propagate quickly. In addition, once cracks are open, they expose the interior of the structure to weathering. This is particularly critical in steel reinforced concrete, which is often used in segmental tunnel linings. In other applications, for example road asphalt, cracking may impact the serviceability state. Cracking is also often aesthetically unwanted. (Munawar et al., 2021) have recently reviewed the different methods proposed for identifying cracks from photographic images. (Mohan and Poobal, 2018) conducted an earlier review that also covers images of data from non-visual sources. While crack detection is not the focus of this study, computer vision methods first applied for crack detection may be adapted to identify other damages. The following sections outline

advances in automated tunnel defect detection methods and discuss their potential application to masonry lined tunnels.

3.2.1. Early studies

Before the advent of machine learning, it was challenging to develop methods that identified the difference between lining features with similar properties such as joints and cracks. Early computer vision methods consisted of image filters and algorithms to highlight particular features. Thresholds on the outputs of these algorithms were used for making conclusions about the nature of the scene in an image. Outside of controlled conditions, however, these methods often generalised poorly when applied to scenes recorded in different conditions and on tasks with more complex and varied situations.

(Ukai et al., 1996) were the first to note the potential for semi-automated visualisation of railway tunnels using computers. They collected continuous scan images of a tunnel lining and highlighted cracks using classical image processing techniques such as dynamic binarisation. Other early crack detection methods, for example (Ukai, 2007), used simple hysteresis thresholding to photographic image brightness levels to identify cracks in concrete lined tunnels. Their method took advantage of concrete-lined tunnels appearing homogeneous aside from cracks which appeared darker due to internal shadows.

(Yu et al., 2007) developed a method to detect concrete tunnel lining cracks from Charged Coupled Device (CCD) cameras mounted on a mobile robot. Sobel edge detection and the Laplacian operator were applied to highlight cracks in the image. A geometric threshold method was used to determine which pixels were parts of cracks and then Depth first search and Dijkstra's

shortest path method were used to extract each crack from the image. (Medina et al., 2017) detected concrete tunnel lining cracks from RGB photographs using rotation-invariant Gabor filters. A genetic algorithm was used to optimise filter parameters. The Gabor filters detect line features well; however, misclassifications occur where other linear features exist such as lining joints or service cabling.

A different approach utilised during materials testing involves taking pictures before and after changes in the material stress state and using digital image correlation (Peters and Ranson, 1982) to detect the formation of cracks from differences in the images. Cracks are defined as areas where detected strain levels are above a set threshold. (Rezaie et al., 2020) noted that this method compares unfavourably to modern CNN based crack detection methods such as their U-Net variant TernausNet. In addition, it is necessary to have pictures of the tunnel lining before the crack formed, which may not be available during inspections of real-world tunnels.

More recent research has also been conducted on tunnel defect detection methods using change detection. (Stent et al., 2016) created a workflow for automatically detecting abnormalities forming in concrete lined tunnels. Their method is invariant to different methods of recording photographic tunnel data during successive tunnel inspections. A trolley with cameras mounted in a semicircular array pointed at the tunnel lining needs to travel along the tunnel during the first inspection. SIFT is applied to generate feature descriptors of the tunnel. Standard SfM photogrammetry techniques are then used to create a 3D point cloud of the tunnel from the photos. Since this method would be applied to cylindrical concrete lined tunnels, quadrics are used to fit a surface to the point cloud. When photos of the tunnel lining are taken during subsequent inspections, their location can be automatically

localised onto the 3D model using k-nearest neighbours matching with the photo's SIFT features and Random Sample Consensus (RANSAC) based registration. Changes are then detected between the initial tunnel state and that in the new photo using various distance functions. The changes are then grouped using unsupervised clustering and ranking, so they can be clearly visualised on the 3D model to the inspector. While it is important to track newly formed defects in historic masonry lined tunnels, it is also necessary to identify the size of existing defects that may threaten the structure's serviceability but have only been recorded in past condition assessments using traditional manual methods. As a result, change detection methods are insufficient to fulfil the comprehensive defect mapping requirements specified within Network Rail standards.

(Dawood et al., 2017) were one of the first to look at segmenting concrete spalling damage on subway tunnel linings. They used RGB images as input to a custom hybrid algorithm that involved multiple stages of image enhancement and filtering, followed by brightness thresholding. They used trial and error optimisation to find the best parameters for their target tunnel. Overall, this method relied on the darker colour of spalled areas compared to the brighter undamaged concrete surface. They used a regression model to find the area of spalling.

Many of these methods work well in relatively featureless concrete lined tunnels when consistent lighting can be achieved. They work particularly well for modern metro tunnels where the whole length has been constructed using a standardised procedure and the level of deterioration is low. Given the length of tunnel on a cohesively constructed metro line may be in the tens of kilometres, there are substantial efficiency benefits to an automated method even when algorithm parameters need to be first manually tuned over a short

tunnel section. However, these methods will perform poorly when faced with the varying tunnel geometries, lighting conditions, material types and larger levels of deterioration present in many masonry lined tunnels.

3.2.2. Machine learning based workflows

(Koch et al., 2015) conducted a review of computer vision methods used for the condition assessment of concrete infrastructure in 2015 and recorded many non-machine learning and a few non-deep machine learning methods with potential application to concrete tunnel linings. Since then, convolutional neural network (CNN) based computer vision methods have been proven to be more effective at replicating human outputs to image analysis tasks. This makes them particularly useful for automating manual visual condition assessment methods. Multiple reviews of the field have been conducted (Deng et al., 2022; Dong and Catbas, 2021; Koch et al., 2015; Sony et al., 2021; Spencer et al., 2019; Ye et al., 2019) given the recent expansion of research interest aided by the recent increased accessibility and effectiveness of computer vision methods using deep learning (Schmidhuber, 2022).

Since 2015, many studies have evaluated either existing or custom CNN designs for condition assessment of concrete lined tunnels. (Makantasis et al., 2015a) used CNNs to detect defects in concrete tunnel linings from pictures of the tunnel lining. Defects were segmented based on an areas texture, frequency, entropy and edge detection output. The algorithm was shown to be the most effective in identifying damaged areas of concrete. It achieved a greater accuracy and F1 score than SVM (Support Vector Machine), Ctrees (Classification trees), KNN (k-nearest neighbour) or ANN (Artificial neural network) methods. Their method produced a relatively noisy

output and identified many small areas of surface concrete mottling and spalling which would not normally be considered defects during a structural inspection. It was therefore difficult to identify the severity of concrete texturing within the locations highlighted by the algorithm.

Initially, CNN architectures were only developed for classification tasks. (Rao et al., 2020) compared the effectiveness of various neural networks at detecting cracks by classifying 64x64 image patches of concrete structures. Each patch was categorised to determine if it contained a crack. Later studies such as Protopapadakis et al. (2019) and Huang et al. (2018) adapted the method to better segment exact damage locations by splitting tunnel lining images up into overlapping patches. Damage 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 to more clearly localise damage locations (Gao et al., 2019). Subsequently, Mask R-CNNs were developed that could semantically segment damage within each boundary box (Xu et al., 2021). Studies have been undertaken to identify the most effective neural network architecture for condition assessment on different structures. (Xu et al., 2022), for example, compared YOLOv3 with faster and Mask R-CNN for detecting and localising cracks in concrete pavements. They showed that the R-CNN methods outperform YOLOv3 when there is limited data of only 150 training images.

Finally, encoder-decoder based networks, such as variants of the U-Net (Feng et al., 2023), alongside 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.

Many of these methods still do not perform adequately when faced with less homogeneous 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)or example, developed FI-SegNet for concrete tunnel lining multi damage detection from photographic data. By combining the existing SegNet with a focal loss function to focus on the most challenging to detect damage examples, they increased the mean intersection over union score (IoU) for crack and spalling segmentation in challenging low-light conditions. (Wang and Shi, 2024) developed a method to address the problems of insufficient lighting and environmental interference in concrete lined tunnels when segmenting water leakage from photographic data. They proposed a customized Side Guided U-Net. Their method integrated Otsu thresholding into an adapted U-Net to encourage the network to learn relationships at different brightness thresholds, helping it to generalise to operate on unseen data with different brightnesses and contrasts than in the training data. Their network achieved an increase in mean pixel accuracy of 3.2% over a baseline U-Net. However, even when machine learning algorithms have been adapted to overcome specific challenges, better results can usually be achieved when baseline models are used with a higher volume (Sun et al., 2017) of clearer (Bergstrom and Messinger, 2022) or more representative (Hu et al., 2021) training data. As a result, there is a substantial benefit to collecting data for training and end user application using standardised methods such as dedicated photographic trolleys or using TLS. Furthermore,

for damages such as spalling or cracking that have an inherently 3D aspect, using the tunnel lining surface morphology may be a more representative and reliable data source than colour image data.

3.2.3. 3D defect detection

While the majority of studies have focused on identifying defects on 2D photographic images of tunnel linings, relatively few studies have developed methods for identifying tunnel lining damages from 3D point cloud data. Segmenting tunnel lining damages directly onto 3D point clouds is beneficial for the end user, as assessors can observe the spatial location of each defect in 3D and more easily communicate damage locations to maintenance staff. (Yoon et al., 2009) conducted early work using geometrical methods to automatically identify concrete tunnel lining defects from lidar data. Defects were defined as points with a significant offset from the plane of the tunnel lining. Using geometry as the classifier ensures that the resulting segmentation has a physical grounding and makes it is easier to explain the causes of false positives. However, using geometry is complicated by other possible causes of abnormal geometries such as from joints and service ducting. In addition, in more heavily damaged tunnels with multiple significant geometrical abnormalities, it is not possible to consistently classify the nature of each damage using thresholding alone.

(Ao et al., 2018) focused on using the surface roughness of the tunnel lining to segment pathologies. They first fit an ellipse to an MLS generated point cloud of a concrete lined tunnel, then removed all inlying points that may be caused by Mechanical and Electrical (M and E) systems inside the tunnel. They then created a mesh from the remaining points and defined the surface roughness as the ratio of the mesh surface area to that of the fitted ellipse.

A larger mesh surface area will occur when the surface is rougher, and the level of surface damage is larger.

Shape fitting algorithms such as RANSAC have been applied to railway tunnel point cloud data. (Sánchez-Rodríguez et al., 2018) applied RANSAC with a Support Vector Machine (SVM) classifier to segment a 3D point cloud of a concrete lined railway tunnel into areas of tunnel lining, ground surface, rails, catenary and cantilever arms. They achieved an average F1 score of 81.15%. (Soilán et al., 2020) later gained superior results using deep learning to segment the tunnel lining, rails, ground and catenary. They compared the Kernel Point Convolution (KPConv) Neural network with PointNet. They achieved a maximum mean F1 score of 92.34% with PointNet and 96.85% with KPConv. This method could form a useful first step to isolate parts of a tunnel obtained by lidar prior to defect detection. While useful for identifying objects within a 3D scene, part segmentation methods are less suitable for directly segmenting damage since lining deterioration can have substantially variable geometries. However, idealised object geometries can be used as an approximation of an undamaged version of a damaged part of a tunnel, enabling change detection to be used to identify damage locations. (Wu et al., 2019) developed a workflow to identify spalled tunnel lining concrete from MLS generated point clouds using shape fitting. They used RANSAC to fit a cylinder to the tunnel, then used the ratio of the true surface area to the projected cylindrical area to identify rough areas. They defined geometric parameters to differentiate spalling locations from bolt hole locations.

More recent studies have applied CNNs to detect tunnel lining damage from lidar obtained point clouds. The majority of studies take advantage of the cylindrical nature of many concrete lined metro tunnels to identify damage using CNNs designed for 2D image data. Virtual images normal to the tunnel

lining can be taken of a tunnel point cloud provided that the tunnel geometry is known or can be easily approximated. This is known as single surface projection, since the tunnel is effectively unrolled and 3D points are projected onto a single 2D raster of the lining.

(Huang et al., 2020a) used single surface projection for segmenting water leakages on shield tunnel linings. In order to flatten the point cloud from 3D to 2D, they fitted an ellipse to the inside of the tunnel lining. Mask R-CNN was then applied to identify wet lining locations from rasters of lidar intensity. The segmentation output was visualised in a 3D mesh of the point cloud by reprojecting the lidar intensity rasters and detected wet locations onto the mesh. (Zhou et al., 2021) used a similar method to segment areas of spalling in concrete lined tunnels from 3D point cloud data. However, they also recorded the offset of each lining point from the fitted ellipse and rasterised this offset into a depth map. The depth offset and point intensity data were input into semantic segmentation neural networks to segment the spalled areas. The outputs from 3 U-Net variants were compared - DepthNet, IntenNet and IDnet.

Few studies have used photogrammetry for tunnel damage segmentation. (Xu and Yang, 2020), for example, collected a comprehensive set of photographs of a concrete tunnel lining from cameras mounted to a trolley pushed down a tunnel. Photogrammetry was used to create a robust 3D B-spline model of the tunnel. A Mask R-CNN was trained on the 2D images to detect and segment cracks in the tunnel lining, whose location was then recorded on the 3D model.

3.3. Automated analysis methods for masonry

Condition monitoring of historic tunnels has received much less research interest than that of newer segmental concrete lined tunnels. No past work could be found extracting defects automatically from 3D models of masonry lined tunnels, however multiple studies have addressed detecting defects in historical masonry structures more generally. Similarly to concrete tunnel defect identification, older studies on automating masonry damage detection focused on traditional computer vision and geometrical methods, while newer studies generally use machine learning, particularly CNN-based approaches.

3.3.1. Direct approaches

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 more challenging than that for concrete structures. However, multiple studies have applied encoder-decoder style neural networks trained on colour photographic data to segment damages in a variety of masonry walls.

(Wang et al., 2018) investigated the importance of different CNN properties when classifying areas of masonry as being intact, spalled, cracked, or containing efflorescence. They considered the AlexNet and GoogLeNet architectures. The network was trained on a 57,780 × 11,400 orthophoto cut into 420x105 and 210x105 images of individual bricks. A sliding window method was found to achieve a greater accuracy of 94.3%. The disadvantage of this is that the window has been manually sized to be the height of a single masonry block, so the method is not readily generalisable to other structures. (N. Wang et al., 2019) later trained a Faster R-CNN on

two 57780x11400 orthophotos to create boundary blocks around masonry blocks exhibiting spalling and efflorescence. They used a sliding window method with 500x500 resolution. The boundary boxes were noted as being more accurate than their previous study, due to the improved region proposal method used by Faster R-CNN. They deployed their model into a mobile app that can provide boundary boxes to photographs taken live in the field and achieved an average precision of 99.9% on efflorescence and 90% on spalling on those images taken using a phone camera.

(Ali et al., 2019) also trained a faster R-CNN method to detect damaged bricks in a 16th century masonry temple. They trained the model on 163 4864x3468 images captured by drone and 303 3024x4032 terrestrially captured images, each split into patches of 500x500. They achieved a mean average precision of 96.5%. This showed improved results over their previous study combining a CNN encoder with SVM for masonry crack detection (Chaiyasarn et al., 2018).

More recent studies have also applied deep learning techniques to detect masonry cracking, spalling and efflorescence. (Hallee et al., 2021) compared the effectiveness of 3 different CNN models with other machine learning techniques for classifying images of cracked masonry. They considered images taken of a wall built and cracked in a lab and images obtained by google image search. They compared the generalisability of algorithms by only training them on either the more standardised laboratory or varied google images. Each image was cropped to 512x512, then downscaled to 100x100 before processing. Their most effective CNN architecture achieved an F1 score of 93.6% on the lab data, while an architecture with more parameters performed the best on the google images with an F1 of 82.4%.

The CNNs performed substantially better than the traditional machine learning models.

(Dais et al., 2021) evaluated 8 different CNNs for classifying masonry cracks on 11491 224 x 224 image patches, taken from 469 varied images of masonry. The images were taken at different angles and field of views of multiple structures and contained various noisy backgrounds. The most effective network was a pretrained MobileNet, which achieved an accuracy of 95.3%. They then compared 24 different encoder-decoder style neural networks for crack segmentation and found the networks with the highest F1 scores to be U-Net-MobileNet, FPN-InceptionV3 and FPN-Mobil, achieving 79.6%, 79.6% and 79.5% respectively. (Asadi Shamsabadi et al., 2022) also examined the feasibility of using neural networks for masonry crack segmentation. They applied a SegFormer model to RGB images. (Karimi et al., 2023) investigated CNN encoder designs for classifying masonry bridge photographs containing 6 different types of damage. Their highest performing network was an Inception-ResNet-2, achieving an accuracy of 96.58%.

(Huang et al., 2024) proposed improving the performance of encoder-decoder based masonry crack segmentation by fusing photographic RGB with thermal camera data. They used a radiation-variation insensitive feature transform to register the infrared and RGB images to create 6 channel RGB-IR images. They then evaluated multiple U-Net style CNN-based and transformer-based neural network architectures for crack segmentation. Operating on the 6 channel images, their best performing network achieved an IoU of 0.73, a 4 percent improvement over using unfused 3 channel RGB images. Their thermal data was collected outside on sunny days when the

thermal contrast is particularly pronounced, so it is unknown how well the method would perform inside tunnels with less clear temperature gradients.

Some studies have investigated detecting masonry damage from 3D data. (Sánchez-Aparicio et al., 2018) created a workflow for identifying defects on historic masonry structures from lidar data using both 3D geometric and radiometric lidar intensity data. They first split the structure into different zones using RANSAC segmentation. Defect identification involved training a CANUPO classifier on the geometric data to identify areas with significant material loss. They used the fuzzy K-means clustering algorithm on the intensity data to identify clusters of similar intensity. After mapping these clusters to identified defects, this method was successful at highlighting areas of moisture, biological colonisation and salt deposits. This approach could be useful to aid human review, however since only filters and clustering-type algorithms are used, the method is unable to identify the nature of each highlighted area without human labelling.

(Enrique Valero et al., 2018) segmented areas of erosion, mechanical damage, delamination and discolouration on ashlar masonry walls from 3D lidar and photogrammetry data using a single surface projection approach. The 3D wall data was first fitted to a 2D plane. The offset of each 3D point from the plane and the RGB data of each point were used for classification. The RGB data was converted to HSV (Hue-Saturation-Value), as this better aligns with human visual perception and so should achieve results closer to those obtained by human structural assessors. Outliers in the HSV and geometrical offset values were defined as areas containing defects. Each area was then classified from a selection of parameters and second order statistics (such as surface roughness, energy intensity and dispersion of hue)

applied to the offset and HSV data. A one vs all logistic regression method was trained on these parameters to classify each defect.

Some studies have applied change detection methods to detect damaged areas in TLS surveys of real-world masonry structures. For example, (Fehér and Török, 2022) used iterative closest point matching on TLS surveys of ruined churches in Hungary to identify recently formed surface damage. They conducted closest point 3D distance calculations between the clouds and classified damaged areas based on the distance between them. (Mouaddib et al., 2024) used a similar method to conduct change detection on photogrammetry obtained point clouds from before and after a fire in the Notre Dame cathedral in Paris, France. They registered the photogrammetric point clouds to a sparse georeferenced point cloud survey of the structure. They used a bespoke manual registration method, followed by closest point 3D distance calculations to identify deformations that occurred in the structure between when the two sets of photographs were taken.

(Zonouz et al., 2023) looked at identifying spalling on individual masonry blocks. They used structure from motion photogrammetry to create a 3D mesh of a block and segment its spalled areas in 3D. They considered the design geometry of a masonry block as being a rectangular prism with no concave features. They then computed the convex hull over the spalled block. Differences between the block’s mesh and its convex hull are spalled areas. However, this method will underestimate the spalling volume when spalling occurs across a whole masonry face so that there are no originally external points on one of the surfaces.

Historic structures typically contain patches of non-original material originating from repair works conducted at different times over the structure's life. As a result, it is useful to first identify the construction materials before

conducting defect detection. (Yuan et al., 2020) trained and compared the effectiveness of different classification algorithms in identifying the construction material of building facades. Facades containing one of ten construction materials were considered. Roughness, HSV and reflectance data was calculated from lidar surveys of each structure and the effectiveness of training on different feature combinations were compared. A support vector machine-based method achieved the highest accuracy.

While most studies have focused on using lidar for 3D damage detection on masonry walls and in buildings, and no studies could be found investigating using lidar for masonry lined tunnel analysis, a few studies have investigated using lidar data for part segmentation of masonry arch bridges. (Riveiro et al., 2016) conducted masonry bridge part segmentation from 3D point cloud data using Principal component analysis and a complex empirically tuned workflow of geometric descriptors. The BridgeNet and BridgeNetv2 neural networks were later developed by (Jing et al., 2022) and (Jing et al., 2024) respectively for part segmentation of masonry bridges. With only 7 real world bridge datasets, the studies used parametrically generated synthetic bridge point clouds for training the neural networks, showing how synthetic data can enable supervised deep learning methods to be applied effectively when collecting extensive real world lidar surveys for model training is impractical.

Overall, most masonry damage detection methods using photographic RGB data have focused on detecting cracking which has a more clearly defined boundary than other types of damages. The best damage segmentation results have been achieved using encoder-decoder style CNNs on photographic RGB data. However, for damage segmentation to be useful in condition assessments, it is vital that the damage severity is classified and the volume of damage can be defined. For masonry spalling, this requires

knowledge of the depth of spalling. This would be obtained more accurately from 3D point clouds obtained by TLS, than those obtained by photogrammetry. Furthermore, using lidar ensures that a comprehensive record of the structure's geometry is collected, which is useful for detecting tunnel deformations when compared against future surveys.

3.3.2. Masonry block segmentation

In masonry lined tunnel 3D point cloud data, dry masonry joints or areas of mortar are recessed from the surface of masonry blocks, breaking up the geometry and making it difficult to manually identify lining abnormalities. It is therefore useful to first semantically segment masonry joint locations before conducting brick damage segmentation. This enables individual block locations to be isolated, allowing analysis to focus either on the blocks themselves or at the boundary between them. Masonry joints complicate defect segmentation, as they can also be easily misidentified as cracks. In addition, mortar loss can be easily confused with masonry spalling. An important part of identifying tunnel defects involves identifying their spatial location on the structure. Being able to easily visualise these locations allows engineers to better analyse the significance of each defect and communicate maintenance instructions to workers on site. In the case of masonry tunnels, automatic labelling of individual masonry blocks would help with this visualisation. Some studies have directly investigated instance segmentation of individual masonry blocks. (Brackenbury et al., 2019) demonstrated that masonry defects are detected more effectively by CNNs when the masonry and mortar areas are separated and processed separately. They trained the GoogleNet Inception v3 architecture, pretrained on the ImageNet dataset, on 94 masonry arch bridge images. A sliding window method was used to identify areas of each image as containing cracking, material loss or

vegetation. A comparison was made between cases when mortar and bricks were separated and processed separately, assigned as different categories in the classification, or not distinguished at all. (Loverdos and Sarhosis, 2022) evaluated multiple encoder-decoder CNNs for both masonry joint and crack segmentation in masonry walls and showed that standard semantic segmentation neural networks can be applied similarly for segmentation of masonry blocks as for segmentation of damage on RGB photographs.

(Idjaton et al., 2021) also compared masonry joint semantic segmentation methods on RGB images. They compared the Segnet and Deeplabv3+ neural networks with the Otsu thresholding and Canny edge detection algorithms. The deep learning based methods achieved substantially better results when the wall photographs were taken under varying lighting conditions, although the Canny method's performance was comparable with the deep learning approaches when the lighting conditions were consistent.

(Kajatin and Nalpantidis, 2021) developed a mortar joint segmentation method designed to enable a mortar raking robot. They created an ensemble method that combined the outputs of 8 machine learning techniques including 1 deep learning technique. The model operated on HSV and depth data obtained from a stereo vision camera. The model was only tested on brick images taken in good lighting conditions from a consistent viewpoint.

(Minh Dang et al., 2022) demonstrated how deep learning based brick instance segmentation and masonry crack segmentation could be integrated into a practical masonry wall condition assessment workflow. They used DeepLabv3+ for crack segmentation and Mask R-CNN for brick instance segmentation from RGB photographic data. They then combined a skeletonization of the crack segmentations with knowledge of the masonry

geometries from the brick segmentation to generate accurate crack length measurements.

Earlier work to segment block locations from RGB masonry images predominantly used traditional computer vision techniques. (Oses et al., 2014) investigated a combination of algorithms such as Canny edge detection and Hough transforms. Full segmentation was found to be challenging without depth data. They note that the low signal to noise ratio of masonry images make conventional edge detection methods difficult to apply uniformly. Their parameters would need to be tuned for each image given different lighting conditions, masonry colours and irregularities within each masonry block. However, by extracting statistical features from the output of image enhancement methods and using them to train a variety of machine learning classifiers, they were able to classify the type of each masonry wall into containing regular blocks, irregular blocks arranged in rows, or irregular blocks not arranged in rows.

A few more recent studies have created masonry segmentation methods that do not require machine learning. (Brackenbury and Dejong, 2018) created a method of mapping individual mortar joints from photographic images using Hough transforms and Sobel edge detection. The method utilised the regularity of joint spacing to separate masonry joints from defects and irregularities in the image. The method required the masonry base course to be horizontally aligned, but only a few parameters needed to be tuned to achieve a good segmentation on walls with relatively consistent geometries.

(Abu-Haifa and Lee, 2022) developed a workflow to use brick wall photographs to automate the generation of discrete element models of masonry buildings that could be used to analyse the structures stability and model their earthquake behaviour. They used the Matlab Image Segmenter

tool which iteratively thresholds the image intensities. It requires rough knowledge of the expected brick geometry to effectively set the segmentation parameters. They then used infilled connected components to remove noise and approximated the detected bricks to simplified rectangular polygons for input to DEM software.

A couple of studies have applied computer vision to segment masonry blocks in 3D point clouds of masonry structures. For stone walls with no mortar, (Sithole, 2008) defined gaps between stone blocks as being locations with points further from the location of the lidar scanner than those on the block surface. Thresholding trial and error was used to segment individual block locations. While performance was acceptable on walls in good condition and with relatively deep and wide mortar channels between the blocks, it did not perform well on the less homogeneous masonry found in most historic structures. (Enrique Valero et al., 2018a) achieved improved performance with a geometric method to identify masonry block locations. They projected 3D point cloud data into a 2D depth map to segment masonry blocks. They then applied an algorithm that utilised 2D continuous Wavelet transforms to segment blocks from the depth map. While their method has the flexibility to deal with surfaces in any orientation and works well on rubble masonry, performance decreases when joints form only small deviations from the masonry block surface, such as found in many brick walls.

(Belén Riveiro et al., 2016) segmented masonry blocks using intensity data obtained by lidar. They used a Multiview approach, rasterising the data onto a best fit plane. Marker assisted watershed segmentation was applied to the intensity data. Filters were used to enhance vertical and horizontal lines that are likely to be mortar joints. The results were strongly dependent on the

clarity of the mortar in the intensity image, so performance was better on masonry with undamaged mortar.

(Shen et al., 2019) conducted brick segmentation on a lab built wall. They used TLS to collect 3D point cloud data of the wall, then used Principal Component Analysis to align the wall with a flat surface. They then applied a custom geometric rectangle fitting algorithm to identify the masonry block positions from the offsets of the point cloud from the flat surface. (Shen et al., 2020) later used the lidar intensity values for brick segmentation. They used k-means to threshold the histogram of lidar intensity and identify the darker mortar areas. Both methods assume similar masonry geometric and colour parameters respectively across different blocks, but work well in a lab setting using homogeneous bricks.

Some studies have noted that the key output of masonry block and joint semantic segmentation is instance segmentation of each block. (Ibrahim et al., 2019) looked at improving the effectiveness of deep learning based brick instance segmentation. They applied a U-Net to segment masonry joint locations and noted that in places where areas of mortar are obscured or adjacent bricks are touching each other without mortar, the U-Net output does not completely separate adjacent bricks. This is because the U-Net is only semantically segmenting the blocks and mortar, and not identifying individual brick instances. They post processed the U-Net output by applying a marker-based watershed algorithm to link up the gaps in the mortar areas to ensure that there is a complete outline around each block. (Vandenabeele et al., 2024) later assessed the real world performance of deep learning-based masonry block instance segmentation on a photogrammetric model of Padua cathedral in Italy. They applied an existing DeepLabV3+ model trained for masonry block segmentation to segment masonry joints before

using a connected components analysis for instance segmentation of individual blocks. The DeepLabV3+ based method achieved better performance than traditional filter-based computer vision using Otsu's method for colour thresholding. However, they also noted that the deep learning method required post processing with a watershed algorithm to achieve consistent masonry joint closure and accurate block instance segmentation.

(Pavoni et al., 2022) took a different approach to masonry block segmentation and proposed Taglab, an AI assisted annotation workflow for masonry block and masonry type and damage segmentation. If unaided deep learning based methods cannot consistently segment masonry block locations, especially in unseen scenarios, then a human in the loop method may be preferable. The study assessed Taglab on photographic data of ancient city walls in Pisa, Italy. Taglab requires an inspector to manually click 4 points within each masonry block. A CNN will then automatically draw a boundary around the block creating an instance segmentation. While much slower than inference of a trained masonry block segmentation neural network, the method provides substantial labour savings over fully manual annotation of each block.

In conclusion, conducting masonry joint segmentation is a useful precursor task to identifying masonry defects that also aids with structural documentation and the communication of damage locations. Many studies have shown that masonry joint segmentation can be effectively achieved using supervised CNN based methods, but there are still barriers preventing its application to real world condition assessment pipelines. While some studies have shown promising joint segmentation performance on unseen test data, it is unknown how well a trained masonry joint segmentation model

would perform on unseen structures with higher damage levels or on those constructed from different materials. In addition, state of the art methods often fail to fully enclose the joint locations around each block, so post processing with a watershed method is required to segment individual block instances. This requires manually tuning watershed parameters on each application. Masonry joint segmentation methods are further analysed in Chapter 6 and the challenge of converting joint segmentation outputs to effective blockwise instance segmentation is addressed in more detail in Chapter 7.

3.4. Industrial developments

With the improved availability of high quality cameras and lidar scanners, alongside improvements to the accessibility of advanced data science algorithms and AI methods (Hussain Rather et al., 2024), it is increasingly feasible to bring these academic methods into industry usage. Given the long term efficiency gains in automating tunnel condition assessment tasks and the commercial benefits of achieving a first mover advantage, a wide variety of companies have begun developing digital tunnel analysis tools. However, none are yet in widespread commercial deployment and development has been slow relative to other sectors. This is partially due to the lack of historical digital data, the wide variety of tunnel types and conditions and the fragmentation of data between individual engineering consultancies and asset managers. The project based nature of civil engineering also leaves little funding for R&D (Brozovsky et al., 2024; Regona et al., 2022).

These challenges are alongside barriers that exist across many traditional sectors (Edmondson, 2024), including:

- A lack of software development and AI skills in the industry
- Concerns around customer data privacy
- An unknown cost of development with an unknown probability of success.
- Resistance to change from both risk-averse directors and employees concerned about Job security
- Ethical and legal concerns involving accountability when an AI tool generates an incorrect or unsafe solution.

Despite these challenges, within the UK, early industry developments in innovative tunnel assessment methods industry were funded by the British government under the Small Business Research Initiative grant (SBRI, 2019). There were three recipients of this grant: DIFCAM evolution by NPL, Inframonit by Railview and New Automated Tunnel Examination System (NATES) by Amey.

(McCormick et al., 2014) developed DIFCAM evolution, which mounts a Laser scanner and high resolution camera array onto an RRV (Road Rail vehicle) to rapidly collect data on the condition of a tunnel's lining as the vehicle drives through it at speeds of up to 1m/s. The speed of inspection enables more regular inspections to be made. Digital image correlation is used to compare the photos of the tunnel lining between inspections. Changes in the lining are automatically identified, showing the development of tunnel disorders. The point clouds generated by the lidar surveys can be easily compared between different inspections, to identify the progression of tunnel deformation (Mouaddib et al., 2024). Both NATES and Inframonit aim to use machine learning to automatically classify tunnel defects. Inframonit

also aims to better utilise GPR data collected by its Railview RRV vehicle. More recently, Arup has commercialised its proprietary LOUP360 tunnel inspection platform (“Loupe360,” 2024). They advertise attaching a 360-degree camera system to either Boston Dynamic’s quadrupedal Spot robot, UAVs or traditional maintenance vehicles to collect visual panoramic images of a concrete tunnel’s lining. They then use machine learning to automatically identify features and defects on the tunnel lining. They state that their method can be up to twice as fast as manual inspections and reduce the number of personnel required per survey from 6 to 2. The Spot robot has also been used to obtain photographic and thermal imagery inside the confined space of power transmission tunnels (UK Power Networks, 2023).

Outside of the UK, there have been similar commercial developments trialling robotic tunnel inspection methods. Drone mounted lidar has been employed in Japan to analyse recently dug tunnel faces (McNabb, 2025) and in the Czech Republic for highway tunnel inspection (Fly4Future Ltd., 2025). On the ground, robot dogs developed by Deep Robotics in China have been applied for power cable inspection in Singapore (Deep Robotics Ltd., n.d.).

Overall, mirroring the state of academic research in the field, there is no commercial technology advertised for automation of masonry lined tunnel condition assessments. While various systems are advertised for automation of concrete tunnel lining inspections, industry adoption is still in its infancy, and such systems have largely only been applied on a trial basis.

3.5. State of the art and future direction for masonry lined tunnel analysis

Masonry tunnel condition assessment is a largely manual process based on the engineering judgement of individual assessors. As a result, introducing

automation would lead to more repeatable and less subjective results in addition to substantial time and cost savings. There has been much recent research applicable to improving the current tunnel structural assessment methodology. Many new technologies have been proposed to assist the collection of tunnel data. However, advances in the data analysis step have the greatest potential to improve results for the lowest cost.

It is possible to automate defect extraction directly from photographic and lidar data that is already collected during structural assessments. Studies have shown that it is practical to detect and localise masonry cracks, spalling and efflorescence from photographic data of masonry walls. Similar studies on concrete lined tunnels have also been conducted that demonstrate the feasibility of automated tunnel damage segmentation using lidar or photographic data. The studies demonstrate that often an approach tailored to a structures geometry and expected damage features is required to achieve state of the art results. It is also possible to detect the tunnel lining construction material, isolate the tunnel lining from a 3D point cloud and segment individual masonry blocks. However, no studies have attempted these tasks on masonry tunnels, and only limited research has been conducted on detecting masonry defects from lidar data. This thesis focuses on segmenting the severity of masonry block spalling and tunnel lining deformation in historic masonry lined tunnels. The majority of past research focuses on only identifying damage locations and does not consider damage severity, even though damage severity is the key indicator of a structure's prognosis. Expanding on existing research in the field, this study targets the following under-researched topics:

- Segmenting the severity of masonry spalling damage is typically a time consuming manual procedure, which no previous studies have

attempted to automate at a full structure scale. This study investigates how deep learning can be used to automate the segmentation of masonry spalling severities from lidar data.

- Identifying masonry tunnel deformations is challenging when historical lidar surveys are not available. There are no existing tools to aid the identification of masonry tunnel lining deformations from 3D point cloud data. This study develops a workflow for removing masonry surface damages from 3D point cloud data, making it easier to identify tunnel lining deformations on the point cloud.
- While multiple studies have applied deep learning for masonry joint segmentation from RGB data and geometric methods have been applied to lidar data, this study forms the first research conducted on segmenting masonry joints from depth map data obtained by lidar.
- Some studies have proposed post processing techniques to generate individual masonry block instance segmentations from joint segmentation outputs. This study investigates how topological methods can be applied to integrate the generation of robust masonry block instance segmentations into the deep learning procedure.
- There have been no comprehensive studies investigating how deep learning based damage detection workflows can be applied in real world structural condition assessments. Through the case study of masonry joint segmentation, this study analyses the generalisation performance of deep learning methods across different structures and methods of assessing the level of uncertainty in the neural network’s predictions.

4. Available Datasets and Resources

This thesis assesses the potential of automated approaches to identify and classify spalling damage and lining deformation in historic masonry lined railway tunnels from 3D point clouds obtained by lidar. Conducting lidar surveys of operational railway tunnels has multiple challenges:

1. Surveys must be conducted during railway closure times, usually at night.
2. To ensure a safe working environment, qualified Controllers of Site Safety are required when working from on the track.
3. Industry standard phase-based lidar equipment costs upwards of £50,000, so is usually only owned by tunnel inspection contractors and specialist surveying companies.

As a result, this study combines data made available from lidar surveys of seven historic masonry lined tunnels with synthetically generated tunnel point clouds. This enables the study to demonstrate how an automated inspection approach could be applied to both real world tunnels and a variety of theoretical tunnel lining situations of which real world data was not available.

4.1. Outline of available data

3D point cloud data from real masonry lined railway tunnels was obtained from two sources. Four 3D tunnel point clouds were provided by Bedi Consulting Ltd of operational British railway tunnels. These were collected in the process of conducting routine condition inspections and the data was made available for this study with the support of Network Rail, the tunnels' infrastructure owner. Two brick lined tunnels and two stone lined tunnels

were available. 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. One of the tunnels has had two lidar surveys taken in different years, enabling the change detection potential of the automated workflow to be analysed. For commercial data protection reasons, names and identifiable location details of all Network Rail owned tunnels are not shown in this study.

A lidar survey of a fifth tunnel, the Brockville tunnel, was obtained through fieldwork conducted in collaboration with Professor Mark Diederichs from Queen's University, Canada. The Brockville tunnel forms a substantial part of the Canadian industrial heritage. It was Canada's first railway tunnel and, now disused by railway traffic, is open to the public as a pedestrian walkway and museum. Refurbished in 2012, the Brockville tunnel contains multiple lining types and sections of original and refurbished lining, making it a valuable dataset for verification of an automated damage detection workflow.

The dimensions of each tunnel and length of available point cloud data are outlined in Table 4.1. All tunnels used within the study were in reasonable states of repair and contained no damages requiring immediate remediation. Spalling of masonry and minor cracking was present in all tunnels in varying quantities. The relative condition of each tunnel is described in Table 4.2. Each structure is described in more detail in section 4.2.

Table 4.1: Overview of the five different tunnels analysed

Tunnel	Asset Owner	Lining type	Cross-section dimensions width (mm) x height (mm)	Total Length (m)	Scanned length (m)	Arch shape
T1 (UK)	Network Rail	Stone	8680x5880	67	19.8	Horseshoe
T2 (UK)	Network Rail	Brick	8470x5660	145	19.9	Circular arch on curved sidewalls
T3 (UK)	Network Rail	Stone	9080x8100	138	19.9	Elliptical
T4 (UK)	Network Rail	Brick	8430X5400	N/A	38.8	Circular
T5 (Brockville, CA)	City of Brockville	Mixed	4500X4930	525	525	Variable

KEY: = Tunnel 1 (Stone) = Tunnel 2 (Brick) = Tunnel 3 (Stone) = Tunnel 4 (Brick) = Tunnel 6 (Mixed)

Table 4.2: Current conditions of the tunnels analysed

Tunnel	Comparative condition notes
T1	Masonry in moderate condition. Some areas infilled with brick.
T2	Originally stone lined, largely relined with brick. Brick in good condition.
T3	Masonry in good condition.
T4	1.5m wide section of crown repaired with sprayed concrete. Brick in good condition.
T6	Varied masonry surface condition, recently refurbished in places. Substantial water ingress and calcite formations.

KEY: = Tunnel 1 (Stone) = Tunnel 2 (Brick) = Tunnel 3 (Stone) = Tunnel 4 (Brick) = Tunnel 6 (Mixed)

4.2. Outline of each structure

Different datasets were used within each chapter's analysis depending on the chapter's goals. The application of each dataset is outlined in Table 4.3.

Table 4.3: Usage of each tunnel point cloud within this thesis

Chapter	Datasets applied	Explanation
Chapter 5 - A proposed workflow for masonry tunnel condition assessment	Synthetic data, T1	Development of a workflow to automatically identify masonry spalling severity and lining deformation. Synthetic data was used to represent different tunnel situations and validate the workflow in idealised scenarios. T1 was used to demonstrate the potential of the workflow applied to real world tunnels.
Chapter 6 – Computer vision for masonry joint semantic segmentation	T1	Investigation into the optimum models to use within the machine learning parts of the workflow. The chapter uses parts of T1 as a case study stone and brick lined tunnel. It focuses on evaluating performance within the same tunnel used for training.
Chapter 7 – Generalisation performance	T1, T2, T3, T4	Evaluation of how well the workflow generalises to different tunnels.
Chapter 8 – Predicting real world performance	T1, T2, T3, T4	Uncertainty quantification of the workflow to aid with performance explainability.
Chapter 9 – Verification and conclusions	T1, T2, T5	Real world application of the workflow and verification that it can identify known defects.

4.2.1. Tunnel 1 (Stone lined)

T1 is located at a shallow depth in an urban area in England and was built in the 1840s using cut and cover. Its portal is shown in Figure 4.1. It is a 67m long dual track tunnel and was constructed in an urban area to accommodate road crossings and buildings. The depth of the crown below ground level is mostly around 1.5m, although the minimum was measured as 0.8m. The tunnel passes predominantly through made ground above lower Lias clay. As much of the tunnel passes underneath buildings, the overburden will be more significant. The height of the tunnel crown is reduced by approximately 0.8m over a short section with particularly shallow cover where a road passes 1.5m above. The tunnel is predominantly lined with blocks of Oolitic limestone, although some areas have been infilled with brick masonry. 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.

Figure 4.1: View of T1 eastern portal

4.2.2. Tunnel 2 (Brick lined)

Built during the 1850s, T2 is a deeper, 145m long dual track tunnel located in a suburban area in England. As shown in Figure 4.2, it passes through a mixture of limestone and mudstone beds to take the railway underneath a hill which is now within a suburban area. The tunnel lining cross-section forms a circular arc on top of curved sidewalls. Historic drawings show a horseshoe shaped profile, although it is unknown whether these are design or as-built drawings. There is a 1 in 115 (0.5 degrees) gradient in the tunnel and slight horizontal curvature. The cover from the crown to surface level varies from 2m to 12m. The lining appears to have been originally constructed with stone but has been largely relined with brick masonry. This masonry is in a relatively good condition with small areas of spalling. A GPR survey was previously conducted within the tunnel that identified up to 1.3m of filled void above the tunnel crown and a single infilled shaft.

Figure 4.2: View of T2 eastern portal, Taken by Sheppard, (n.d.)

4.2.3. Tunnel 3 (Stone lined)

T3 is a 138m long, slightly curved accommodation tunnel that carries a road over a dual track railway within a suburban area in England. It was opened in 1840 and passes through sandstone. Tunnel 3 has an elliptical profile that changes to a round arch on vertical sidewalls at the portals as shown in Figure 4.3. Historical photos suggest that it was originally partially unlined and only later lined with brick near the portals according to drawings from the early 20th century. A section of the tunnel consisting of an area of stone lining at least older than 1880 was selected for this study. The overburden varies from 3m to 21m.

Figure 4.3: T3 eastern portal. Photo taken by Moore (n.d.)

4.2.4. Tunnel 4 (Brick lined)

For commercial reasons, an outline and photograph of Tunnel 4's situation is not provided. The tunnel has an elliptical profile over a dual track railway. It is lined with brick masonry generally in a good condition with small areas of low severity spalling. An approximately 1m wide section of the crown is lined throughout the tunnel length with shotcrete. Shotcrete is a sprayed concrete product that is often used for lining repairs due to it's ease of application in different orientations. It is possible that there had been lining deformations. A common deformation involves the tunnel arch flattening with the sidewalls moving outwards and the crown sagging. This is caused by either an increase in loading at the crown or a decrease in lateral ground pressure and will lead to open joints at the crown. It is likely that the shotcrete was applied to prevent brick loss at the crown and the risk to tunnel users from falling masonry blocks.

4.2.5. Tunnel 5 (Mixed lining type)

The Brockville tunnel (T5) is a single track railway tunnel located in Brockville, Ontario in Canada. The tunnel is 525m long and passes underneath the urban centre of Brockville connecting the mainline railways at Brockville station with the waterfront on the St Lawrence River. Its original purpose was to transport lumber from the Ottawa region to the Brockville port from where it could be exported to Europe or the USA. Now owned by the city of Brockville and, with the railway removed in the 1970s, it was reopened to the public in August 2017 as a museum and pedestrian walkway following substantial refurbishment works. It was originally opened by the Brockville & Ottawa Railway Company in 1860. Construction lasted 6 years due to a combination of financial and construction challenges.

The tunnel passes predominantly through clay, with a central 142m section passing through Quartzite rock. It was constructed using multiple methods, with the central section excavated with gunpowder blasting and left unlined. The rest of the tunnel is lined with a mixture of stone and brick masonry. As an English contractor, D. Booth and Sons was chosen, it is likely that the masonry lining was constructed using the English method. At the boundary between the clay and rock areas, there are multiple short masonry lined sections alternating with exposed rock. The view from an area of brick masonry at the south end of the tunnel is shown in Figure 4.4.

Prior to the tunnel's refurbishment, there were multiple plans for future utilisation of the then disused tunnel. Three condition assessment reports have been conducted (Carer and Diederichs, 2012; Grant, 1978; Roberts, 1974). The 1974 report provides a brief overview of the tunnel's history and present condition. The 1978 report provides a more in depth condition overview alongside proposals for future usage of the tunnel. The recent history of the tunnel is also well documented. The extensive 2012 report was conducted to determine the works required to enable the reopening of the tunnel as a pedestrian route. It contains a detailed condition assessment of both the masonry lined and exposed rock sections of the tunnel aided by a lidar survey. This lidar survey follows the approach proposed in (Fekete et al., 2010) and forms one of the first comprehensively generated 3D point clouds of a tunnel lining. The 3D point cloud generated of the exposed rock section of the tunnel has since been used as case study data for studies investigating using Discrete Fracture Networks to model rock masses (Farahmand et al., 2018; Vazaios et al., 2017). Approximately 50% of the tunnel's length was resurveyed by lidar in 2025, enabling accurate ground truthing of any recent changes in the lining's condition. There is a wide variety

of masonry types and conditions within the tunnel. Local conditions are explained in more detail at the assessment area chosen in Chapter 9.

Figure 4.4: View inside of the Brockville tunnel

4.3. Data preprocessing

Terrestrial tunnel point cloud surveys are conducted by placing a TLS system at different locations within a tunnel before combining each cloud in a registration process. As the lidar scanner rotates at a constant angular speed and records each point over a set time interval, there is an even angular density of recorded points around the scanner location. However, the density of points on each surface in the generated point cloud will vary proportionally to the distance of the surface from the lidar scanner and the angle of the surface relative to the angle of incidence of each laser pulse. Surfaces perpendicular to the incident laser will achieve a higher point density than those at more oblique angles. As a result, points circumferentially around the

tunnel's arch will have a relatively even density. However, the point density rapidly reduces along the tunnel's length since both the obliquity of the lining surface and distance from the scanner are increasing. Scans are therefore typically conducted at set chainage intervals along the tunnel's length and then registered to avoid excessive scan times to capture points far from the scanner location.

Multiple factors affect the original point density of the lidar scan. This includes different scanners being used and different scan speeds being applied based on the length of time available on site for the survey to take place. For surveys using modern phase-based scanners, the key constraint is usually the available workhours within the tunnel and there is a trade off between the level of detail of each scan and the chainage interval used between scans. More scans result in more downtime as the TLS equipment is moved and recalibrated each time, but a smaller scan spacing requires less scanning time overall due to each scan only requiring data from surfaces that are closer and at a less oblique angle to the incident laser.

All of the tunnel point clouds utilised within this study were specified with a maximum point spacing of 4mm. Details of the exact procedures used by each surveying company to generate the point clouds provided by Bedi Consulting are not available. However, for the 2025 survey of Tunnel 5, the TLS system used a 15 minute level of detail for each scan and a maximum chainage interval between scans of 5m was used.

Registering point clouds involves aligning them and calculating their relative orientations. Registration can be conducted manually by point matching. However, this method will result in a registration error proportional to the point spacing, as subsequent lidar scans do not record points at exactly the same locations. This method is also prone to human error. As a result,

multiple algorithms have been developed to aid the registration process (Cheng et al., 2018) and many are integrated into registration software. Point clouds obtained from Bedi consulting were registered using the proprietary software bundled with each scanner. Point clouds from T6 were registered using the Autodesk Recap software (“ReCap Pro | Autodesk,” n.d.). If substantial registration errors were visually identified, then the registration process was repeated. Nevertheless, small registration errors do manifest as noise in the combined registered point clouds.

After registration, the generated 3D point clouds were spatially downsampled to a 4mm point spacing before analysis was conducted. The downsampling strategy used iterates through each point in the cloud and either adds it to a new cloud or discards it if it is within 4mm radius of any point added to the new cloud. Subsampling reduces the computational cost of manipulating the point cloud by removing points recorded at an unnecessarily high density directly above the scanner caused by the nature of rotation used by the scanner when recording. The data collection method of a typical phase based terrestrial lidar system involves the scanner rotating slowly around the vertical axis and quickly around it’s horizontal axis. With this method, arcs of points are recorded over the scanner that intersect vertically above it, leading to an uneven default point density. Standardising the point density by spatially downsampling the collected point clouds enables a better comparison between different tunnels, better controlling for the impact of varying point density on performance than the less computationally intensive random sampling method. The number of points in each point cloud analysed within this study is shown in Table 4.4. The raw point cloud data for tunnels T1, T2 and T3 were not available for this study. For a single surface with relatively small surface deviations, the average 3D point density on the

surface is inversely proportional to the square of the average point spacing, although a higher surface roughness will increase the 3D point density. The typical masonry tunnel lining largely follows a single surface with only small amounts of out of plane deviation forming the tunnels curvature, deviations from defects and at the masonry joint locations. The impact of subsampling on tunnel T4 is shown in Figure 4.5 where the high density of points in the raw scan data directly above the scanner can be observed relative to that in the subsampled cloud.

Table 4.4: Size of the tunnel lining point clouds analysed

Tunnel	Raw scan Initial number of points	Subsampled cloud (4mm spacing) number of points
T1	N/A	17,439,453
T2	N/A	15,622,450
T3	N/A	18,797,398
T4	148,868,570	5,105,024
T5	Specific sections outlined in Chapter 9	

KEY: = Tunnel 1 (Stone) = Tunnel 2 (Brick) = Tunnel 3 (Stone) = Tunnel 4 (Brick) = Tunnel 6 (Mixed)

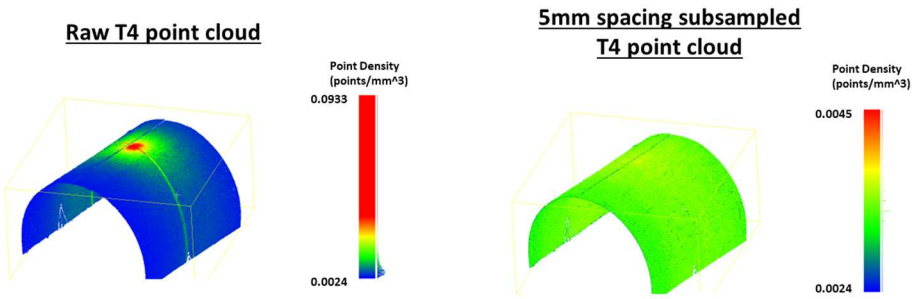


Figure 4.5: T4 point cloud before (left) and after (right) spatial subsampling

4.4. Software and Compute resources

Five different compute resources were utilised in different parts of the project as more powerful computational resources became available. A comparison of these resources is outlined in Table 4.5. The results section of each chapter references the resource used within that section. The ARC4 and AIRE HPC (High Performance Computing) services were provided for the project by the University of Leeds. All code was developed for the project in Python and deep learning was conducted using the PyTorch library (Paszke et al., 2017).

Table 4.5: Compute devices used within this thesis

Name	Type	OS	CPU	RAM	GPU	GPU VRAM	GPU theoretical FP32 performance
PC1	Workstation	Linux Ubuntu 24.04	Intel I7 2600k	16GB	Nvidia GTX 970	4GB	3.92 TFLOPS
ARC4	Hpc	Custom Linux	Intel Xeon 6138	48GB	Nvidia V100	16GB	14.13 TFLOPS
AIRE	Hpc	Custom Linux	AMD EPYC 9254	64GB	Nvidia L40S	48GB	91.61 TFLOPS
L1	Laptop	Microsoft Windows 11	AMD Ryzen9 5980HS	32GB	Nvidia GTX 1650 MaxQ	4GB	2.55 TFLOPS
L2	Laptop	Linux Ubuntu on WSL	AMD Ryzen9 5980HS	32GB	Nvidia RTX 4090 Mobile	16GB	35.31 TFLOPS

5. A proposed workflow for masonry tunnel condition assessment*

**This chapter includes parts of published works: (Smith et al., 2024)*

There are multiple challenges involved when developing automated methods for structural damage segmentation. The non-homogeneous nature of the masonry surface in Victorian-era tunnels makes it particularly difficult to develop a method that generalises well to the wide variety of masonry types and damage levels present in real world structures. As a result, it is also challenging to validate a developed analysis workflow. This chapter proposes an automated workflow for masonry spalling severity segmentation that operates directly on a 3D point cloud obtained by terrestrial lidar surveys. The workflow automatically generates masonry spalling severity maps for direct usage in tunnel condition assessment reports. It also provides lining profile abnormality visualisations that help tunnel inspectors to identify lining bulges and deformations. The study assesses the potential of the workflow on multiple expected tunnel geometries and damage types by evaluating it on synthetically generated point clouds representing different expected tunnel conditions.

5.1. Challenges of analysis

Developing a workflow for automatically analyzing masonry lined tunnel 3D point clouds brings multiple challenges. The properties of 3D point clouds make them more difficult to design algorithms for than 2D images due to four additional complications:

1. **Point clouds are unstructured**. Point clouds consist of a list of point co-ordinates, so spatial relationships between points are initially unknown. As a result, a method must be able to identify and group similar features of an object at varying orientations and scales. This often means that computationally expensive nearest neighbor calculations must be conducted to characterize spatial relationships before further analysis can be conducted.
2. **Clouds can be sparse**. The density of points within a cloud varies spatially due to the terrestrial laser scanning procedure and any post-processing conducted. An algorithm must be able to reach the same conclusions when objects have different point densities or there is a varying point density within each object. There must also be an efficient way of grouping points when there are substantially varying point densities across a cloud. Furthermore, even when the same point density is achieved, exact point locations on a surface will vary between clouds.
3. **Points are unordered**. As a result, algorithm outputs need to be invariant to the order of points within the point cloud, as the same data may generate point clouds with a different point ordering.
4. **3D data processing is computationally expensive**. Transforming 3D data into 2D representations where possible has the potential to bring substantial performance benefits.

In addition to the challenge of working with 3D point cloud data, the extent of damage typically present on many masonry lined tunnels and the wide variety of possible tunnel geometries and masonry patterns poses a challenge for generating repeatable algorithms beyond that tackled in previous studies. In general, tunnels constructed around the same time on

the same line will have been constructed using similar geometries and materials. However, the nature of deterioration and resulting historical maintenance work often varies both between tunnels and within longer tunnels. It is important to ensure that any developed methods are validated over a range of lining situations and that a method is only applied in practice when a target tunnel is shown to have equivalent features to those in locations where the method has been validated.

Typical masonry tunnel lining situations are shown in Figure 5.1. In many tunnels, it may not be possible to use the presence of abnormalities to identify changes from the as built state, since the extent of damage is too high to easily identify undamaged baseline properties. Any workflow developed will need to be flexible to different masonry block geometries and be able to differentiate masonry damage from mortar areas given different mortar dimensions and properties. This can be difficult when mortar areas are very thin or their surface is level with that of the masonry blocks so that there is little geometric relief between them. For identifying masonry spalling, it is also important to differentiate between spalling damage and other surface variations such as efflorescence, vegetation growth or complete brick loss. As a result, analysis will need to be tailored to the specific damage targeted and the limits of the analysis will need to be determined before any developed workflow can be used in practical tunnel condition assessments.

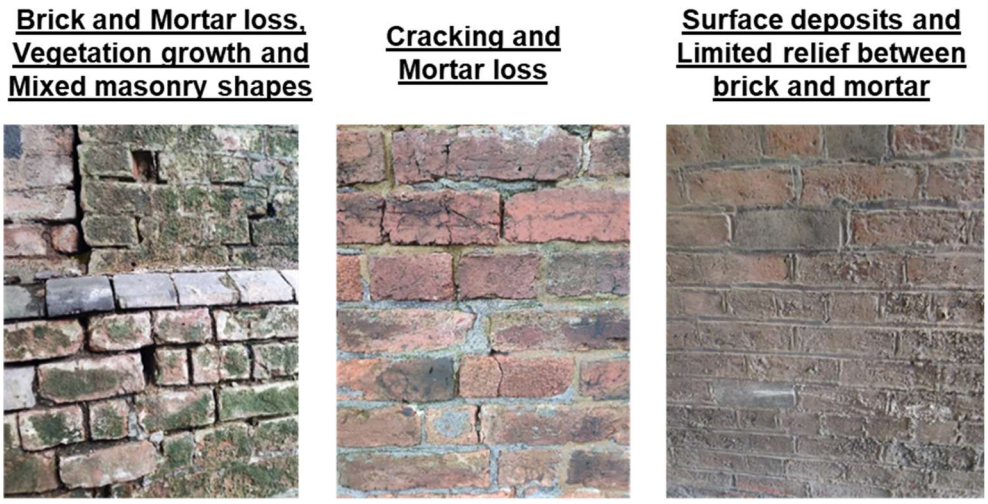


Figure 5.1: Typical masonry tunnel lining complexities

5.2. Outline of proposed workflow

The proposed workflow automatically locates areas of masonry spalling and classifies them into severity categories based on those defined within the Network Rail standard NR/CIV/L3/006/4C (Network Rail, 2016). It then visualises these locations onto the input 3D point cloud. The workflow replaces the manual spalling labelling and severity categorisation steps of the condition assessment process shown in Figure 5.2 with those shown in green within Figure 5.3. Figure 5.3 also shows the 5 stages of the proposed automated workflow.

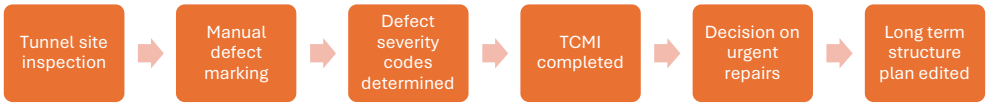


Figure 5.2: Traditional masonry tunnel condition assessment procedure.

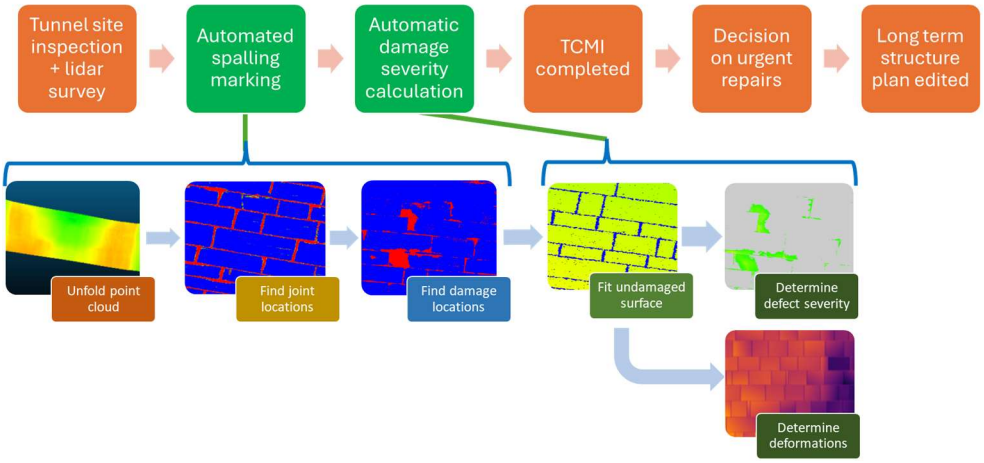


Figure 5.3: Proposed automated workflow within overall condition assessment procedure.

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 10m 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 machine learning. The most appropriate methods for this step are discussed in Chapter 6. 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 machine learning model 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 consists of 5 stages:

1. Align and unfold the tunnel 3D point cloud, then rasterise point cloud parameters onto a 2D image
2. Use machine learning to identify masonry joint locations and isolate individual blocks
3. Use machine 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 directly 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 closures. 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 inspectors. The modularity of the workflow enables future developments in masonry damage 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.

The workflow has been developed such that it can be run on a mid-range Windows 11 laptop and was evaluated on device L1 (see section 4.6). For an industry standard terrestrial point cloud survey subsampled to 4mm point spacings, as shown in the provided benchmarking data outlined in Chapter 4, a typical 20m section of dual track tunnel will yield around 20 million points.

In order to meet the performance requirements without further data loss through subsampling, it is assumed that input tunnel data can be split into maximum 20m long sections before input into the workflow. A detailed performance analysis is provided in Chapter 9.

5.3. Ground Truthing

It is necessary to manually identify ground truth spalling severity maps before the workflow is developed. This will provide both a qualitative comparison of the effectiveness of the automated method with the manual method and a reference for calculating performance metrics of the developed workflow. As shown in section 2.3.3, the manual method applied in industry for calculating spalling severity from 3D point cloud data involves manually defining a 3D surface representing the undamaged masonry surface level. It would be unreasonably time consuming to conduct this on a block by block basis, so the undamaged surface is defined by fitting a curve to the tunnel cross section. For some tunnels, a simple polynomial fit may work well, but for most, manual adjustment of multiple parametric curves connected together piecewise is needed. The curve is extended along the tunnel into a surface. The surface then needs to be manually adjusted locally to account for local tunnel bulging. Once a good fit has been achieved, the closest distance from each point in the point cloud to the created surface is calculated. This distance is saved to each point as a scalar field and is visualized on the 3D point cloud of T1 in Figure 5.4. Masonry surface damage such as spalling and cracking is defined as areas where the tunnel point cloud is outside of the created surface, while points beneath the surface denote surface deposits such as efflorescence or vegetation. Mortar areas are manually masked out afterwards.

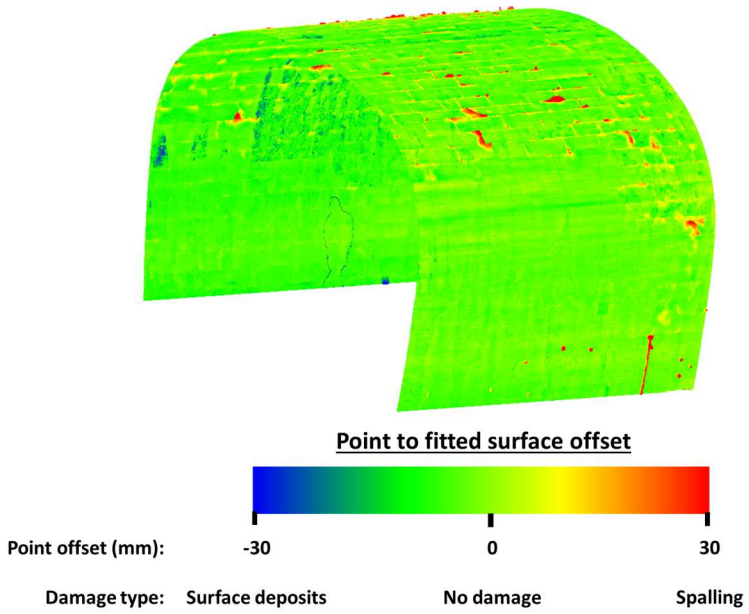


Figure 5.4: Ground truth damage depths on a section of tunnel T1

5.4. Synthetic dataset

Synthetic point cloud data representing different damaged masonry lined tunnels were generated to verify the proposed assessment workflow. The geometries of 20m long example tunnel sections were parametrically generated and their properties were randomly varied within realistic ranges. Realistic upper and lower bounds to the randomised parameters were obtained through trial and error with comparison to the tunnel T1 for stone masonry and T2 for brick masonry. The geometry of each example consists of vertical sidewalls connected with an arch. Initially set as circular, the geometry of the arch was adjusted using vertical scaling to form different profiles. The synthetic point cloud was then generated from the selected geometry by selecting point co-ordinates at 4mm spacings.

Once the base point cloud had been created, damages and abnormalities were added to the cloud. These were generated as images representing

depth maps to be wrapped over the lining and applied as out of plane deviations to the existing point cloud geometry. Depth maps were generated as 16bit .tiff files and pixel values were set as the depth of the feature in metres. All parameters used to create the depth maps were randomised within bounds representing realistic lining features. Features of the lining were added as follows:

- **Masonry Joints.** The block geometry was randomly created bounded within typical stone and brick masonry shapes and block length to width ratios. The joint depth profile was modelled as a normal distribution. Horizontal joints were placed first, followed by vertical joints between them. English and Longitudinal masonry bonding patterns were each given a 35% chance of being used. Otherwise, the location of vertical masonry joints were randomised within the range of standard masonry block sizes.
- **Masonry spalling.** Areas of spalling were added to the blocks using the method outlined in Figure 5.5. Randomly placed control points were used to generate Bezier curves forming enclosed shapes that delineate the area of damage. Often used for modelling naturally occurring geometries (Chi et al., 2003), Bezier curves are parametric curves defined by discrete control points. They were chosen since they are easy to randomise through moving the control points, while still generating enclosed shapes without substantial computational cost. Within an area of spalling, the depth of spalling was first modelled as having a randomised uniform gradient that is generated by a distance transform within the boundary of the spalling area. The maximum value of the distance transform was then scaled to generate spalling with a realistic depth. Further deviations were

added by introducing flat small areas with random offsets. These were smoothed into the image, by applying a Gaussian blur. Finally, the spalling maps were cropped and added to the masonry joint depth map. Spalling was curtailed at the joint locations to ensure that it was contained within each masonry block.

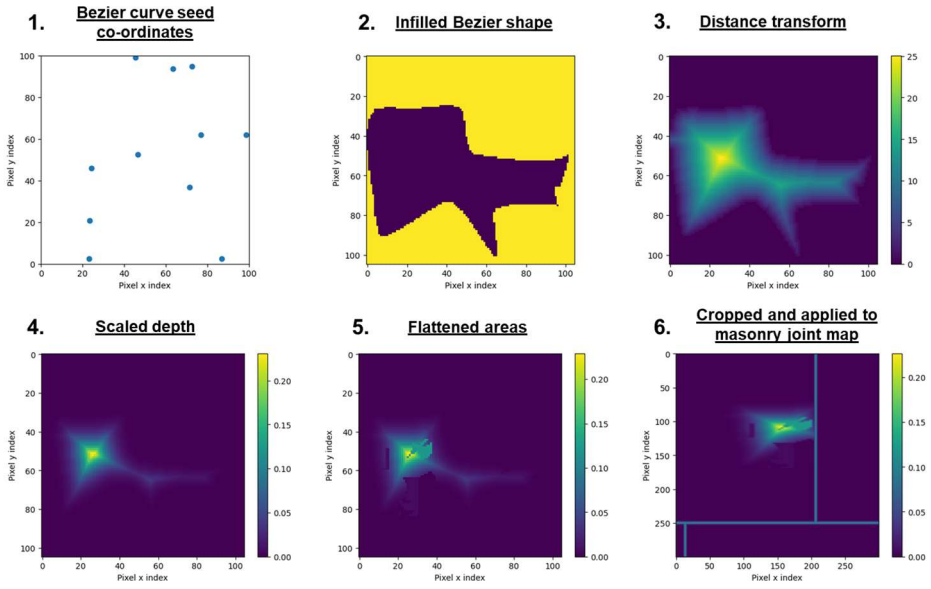


Figure 5.5: Steps to generate spalled masonry depth map

- **Efflorescence and surface deposits.** The shape of effloresced areas was modelled similarly to spalling using Bezier curves. The depth of the deposits were also modelled using a distance transform. However, this was then multiplied with small scale Perlin noise to generate the typical lumpy texture of efflorescence. Perlin noise (Perlin, 1985) was developed for generating pseudorandom computer graphics textures. It has a more organic appearance than standard noise functions and tuneable levels of smoothness. The surface deposits were applied equally to the masonry block and joint areas. The deposit depth maps was subtracted from the overall depth map image in order to form inwards geometrical deviations on a tunnel lining point cloud.

- **Lining deformations.** Large scale Perlin noise was applied to model larger bulges and distortions of the lining.
- **Surface roughness.** Low magnitude, small scale Perlin noise was combined with gaussian noise and applied throughout the lining. These model the masonry's surface roughness, data collection noise arising from the precision limit of a lidar scanner and any scan registration error.

Each of the added depth maps are visualised in Figure 5.6 alongside the overall depth map created by summation of the other depth maps. At each stage, a binary mask of the location of each added feature was saved to use as the ground truth location when testing the workflow. The range of parameters used are outlined in Table 5.1. A substantially wider variety of features including safety alcoves, vegetation and electrical service ducts are often present in masonry tunnels that impact the geometry of the surface observed in recorded point cloud data and obscure or interrupt the masonry surface. In these cases, it is not possible to determine the masonry condition without manually examining behind or inside these features. However, compared to the task of manual spalling segmentation, it is relatively straightforward to manually crop these larger areas out of the input point cloud or use a separate machine learning model to isolate the lining before analysis (Kang et al., 2023). As a result, this study focuses only on each tunnel's masonry lining and assumes that lining point clouds have been fully cleaned of obstructions before input into the workflow.

10 randomised synthetic point clouds of 10m length were generated and run through the workflow in order to visualise each stage. Three of the generated tunnels are visualised in Figure 5.7. Figure 5.8 shows a small section of generated point cloud where areas of spalling can be seen.

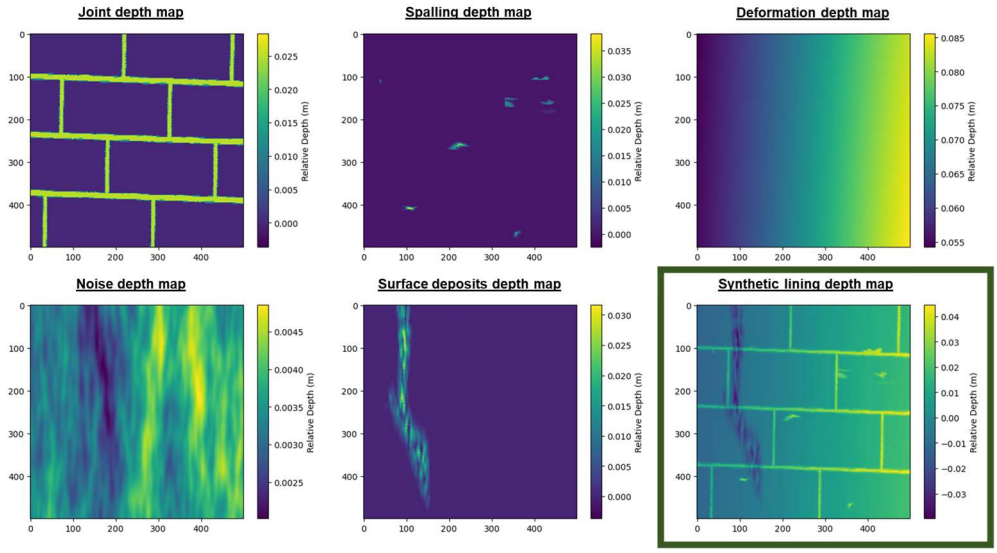


Figure 5.6: Visualisation of depth maps used to generate damaged lining. Outlined in green, the Synthetic lining depth map shows the final combined lining which is formed from a summation of the other depth maps shown.

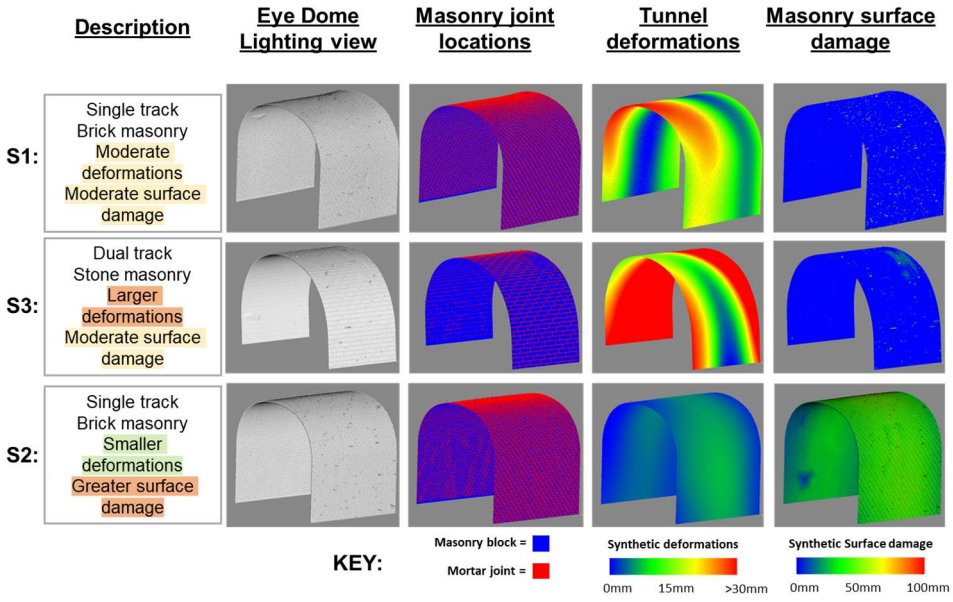


Figure 5.7: Synthetically generated 3D point clouds showing different lining situations. A scalar field was applied to points in each cloud to visualise the location of masonry joint locations, tunnel deformations and total surface damage. Eye dome lighting applies a shader to mimic the surface appearance under natural lighting and highlight surface variations.

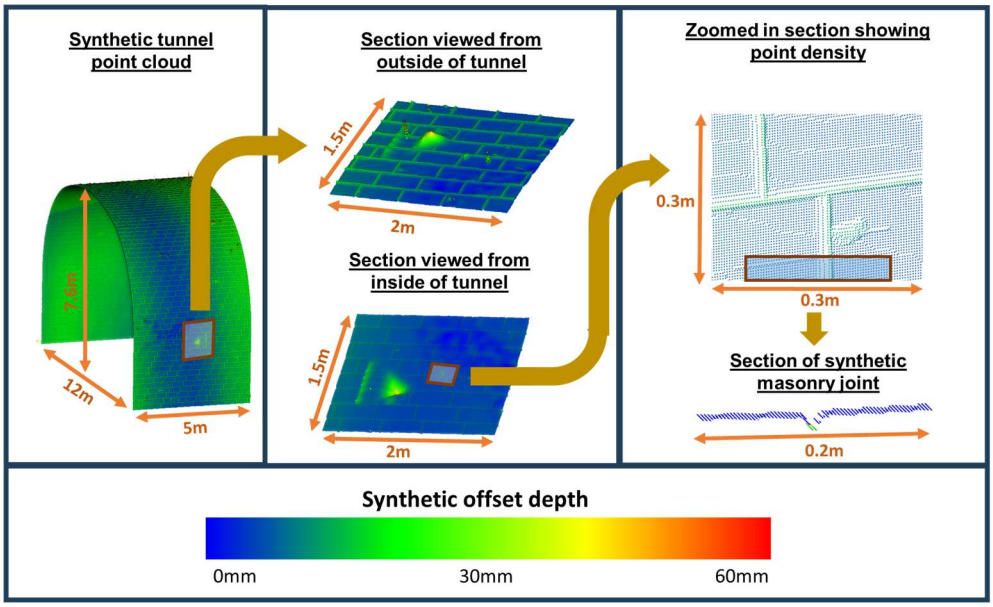


Figure 5.8: Visualisation of sections of synthetically generated tunnel 3D point cloud, with each point's scalar field set as the total damage depth map. Zoomed in sections are shown, highlighting the point density of the generated cloud.

Table 5.1: Synthetic data parameter limits

Stage	Description	Parameter	Lower Bound	Upper Bound
1	Block length	Number of pixels	50 (equating to 0.2m)	250 (equating to 1m)
1	Block height	Number of pixels	Block_length/2.6	Block_length/1.3
2	Horizontal offset between	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

5.5. Proposed methodology

This section describes each step of the proposed spalling analysis workflow. The proposed workflow was implemented using Python. Numpy (Harris et al., 2020) was used to store and manipulate the point cloud and image data. The Laspy package (Brown and Montaigu, 2024) was used to handle 3D point cloud IO, while OpenCV (Bradski, 2000) and Skimage (Van Der Walt

et al., 2014) were used for image processing. Open3D was used to aid with point cloud manipulation (Q.-Y. Zhou et al., 2018). Two further packages were used to reduce the workflow computation time. CuPy (Tokui, 2024), a package which enables GPU accelerated python, and Numba (Numba, 2024), which compiles python code to produce more efficient functions.

5.5.1. Aligning and unwrapping

The 3D point cloud must first be transformed 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 rasterization 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 is first split into sections where the alignment deviates a maximum of 30 degrees from straight. For most standard gauge railways, a maximum length of 20m will ensure that the deviation limit is not exceeded.

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 Figure 5.9. 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 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.

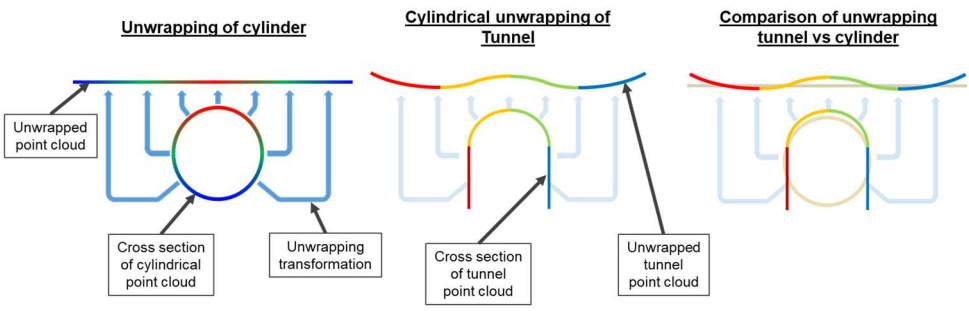


Figure 5.9: Cross section visualisation of the cylindrical projection approach applied to a cylindrical point cloud and that of a typical tunnel's geometry.

5.5.2. Rasterising

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 or other semantic segmentation algorithm chosen for masonry joint and damage segmentation, although the workflow was developed such that lidar RGB, Intensity or other point cloud scalar fields could also be used to aid analysis. The work conducted within Smith et al. (2023) showed a generative adversarial segmentation network achieving the best segmentation performance. Despite this, the single channel defect depth parameter (depth offset) is used here, as it was the best performing single channel method when joint segmentation was conducted using a CNN. If neural networks are used for joint and damage segmentation, then 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. As a result, this study only considers single channel depth map images in the proposed workflow.

Each point offset value or other scalar field was projected vertically downwards onto a 2D plane and rasterised onto a 2D, 16bit half-precision floating-point .tiff format image. For the point offset value, rasterization creates a depth map of the tunnel lining as shown in Figure 5.10. Depending on the geometry of the input tunnel data and level of surface damage, the range of the point offset scalar field will vary. The offset values are scaled between 0 and 1. Each pixel within the 16bit raster has 215 = 65,536 possible values. The uniform quantization error bound states that a quantisation range of double a values accuracy is required to avoid discretisation errors (Gray and Neuhoff, 1998). Assuming that the cloud to be rastered has a 3D point accuracy of no less than 1mm, then using 16bit values will ensure that no

information is lost since a deformation of 1m would require only 2000 unique values.

The resolution of the rasterised image can be manually set, although a minimum best case resolution 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 pixel binning procedure is visualised in Figure 5.11. The transformation from point cloud to image is recorded so that output images can be accurately projected back onto the initial point cloud to aid visualisation.

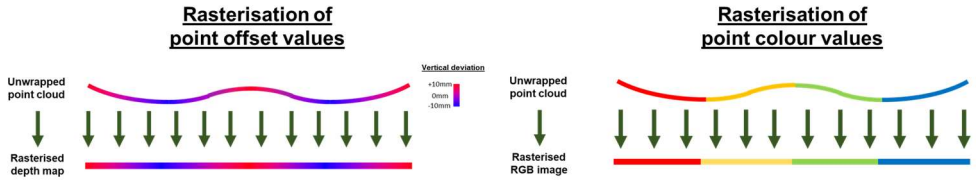


Figure 5.10: Cross section visualisation of the rasterisation approach used to generate lining depth maps and rasters of point cloud scalar fields.

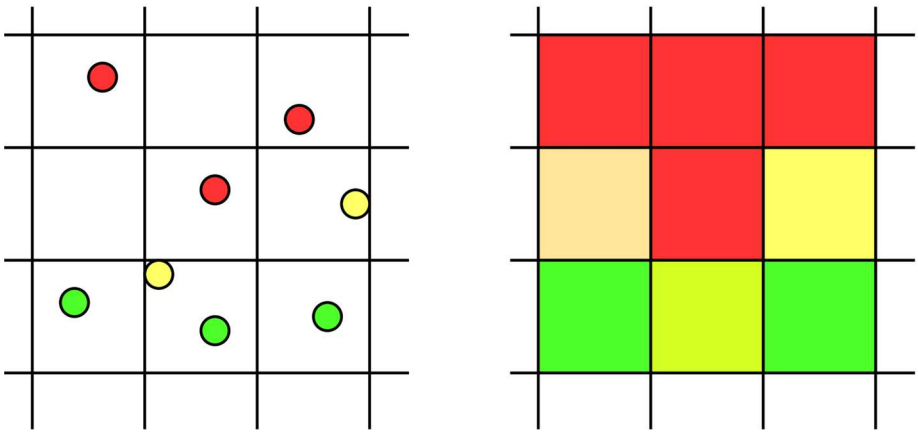


Figure 5.11: Rasterisation example onto a 3x3 resolution image.

5.5.3. Damage and Joint segmentation

Two stages of semantic segmentation are required. First, the binary segmentation of masonry joints and masonry block locations is conducted, then damaged and intact masonry locations are segmented from each other. Both tasks can be conducted from the rasterised depth map images. Determining the ground truth for damage segmentation is more subjective than that for masonry joint segmentation. However, the complexity of the task is reduced since masonry joints may be removed from the images before segmentation so that any applied algorithm can focus on segmenting just masonry surface abnormalities. Segmentation of the undamaged locations aims to remove all types of damage and any objects obscuring the masonry surface from the images, so a segmentation algorithm targeting specific damage types is not required. The created damage location mask is used only to identify which points are not on the original surface plane of each masonry block, rather than for directly determining spalling locations. As a result, a small amount of oversegmentation of the damaged area will not impact the final spalling severity segmentation. While it is possible to manually draw both of these segmentations, and annotating the 2D raster images is considerably more straight forward than annotating the 3D point cloud directly, it would still be time consuming given the large number of masonry blocks on a typical tunnel’s lining. Chapter 6 analyses multiple automated methods for conducting these segmentation tasks.

The aim of the masonry joint segmentation is to identify individual block locations, so it is important that the generated joint mask fully encloses each masonry block. A connected components analysis was used to isolate each individual block instance. The 8-way connectivity Spaghetti labelling algorithm developed by Bolelli et al. (2020) was implemented using the OpenCV library. The output instance segmentation is visualised in Figure 5.12.

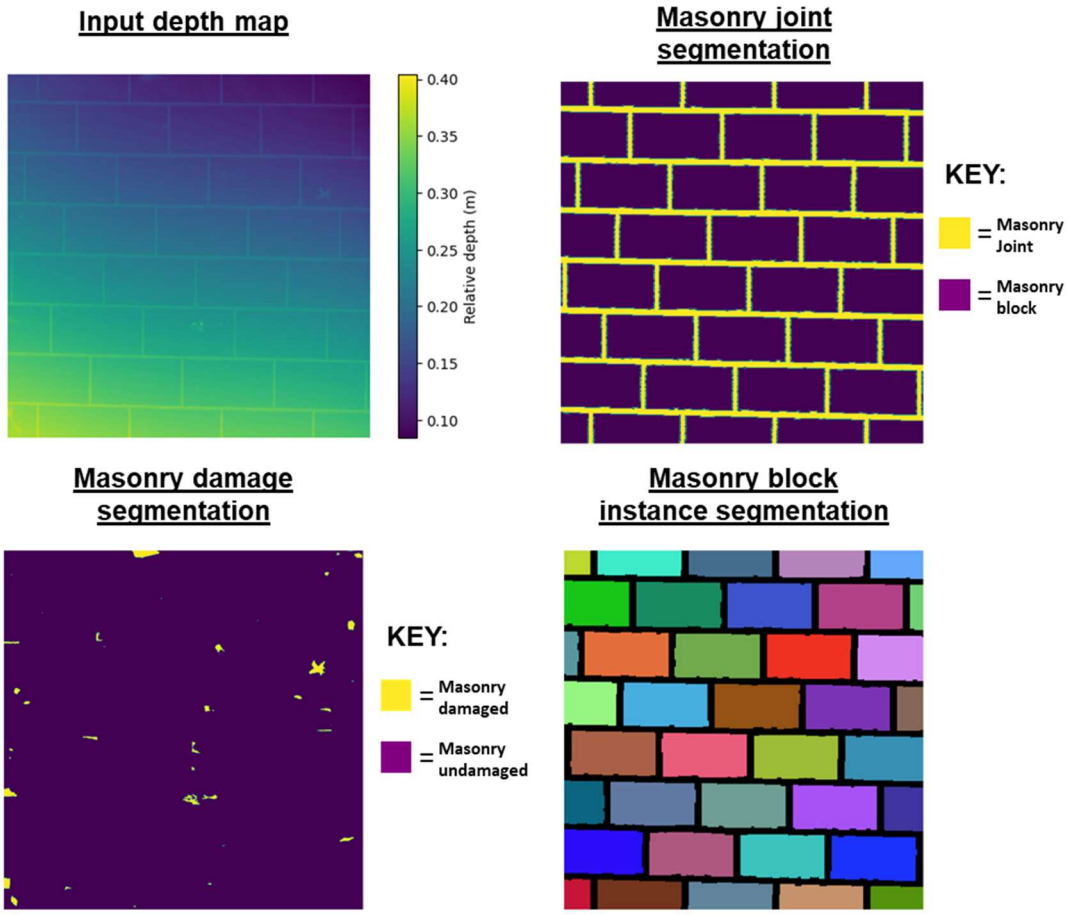


Figure 5.12: Masonry damage segmentation and block instance segmentation outputs on synthetic data.

5.5.4. Plane fitting and Severity Segmentation

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 Figure 5.13:

1. All pixels representing areas of masonry joints or lining damage are removed from the raster image generated in section 5.5.2, using the masks generated by the algorithms developed for section 5.5.3.
2. Using the pixel coordinates of each block calculated in section 5.5.3, 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. 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

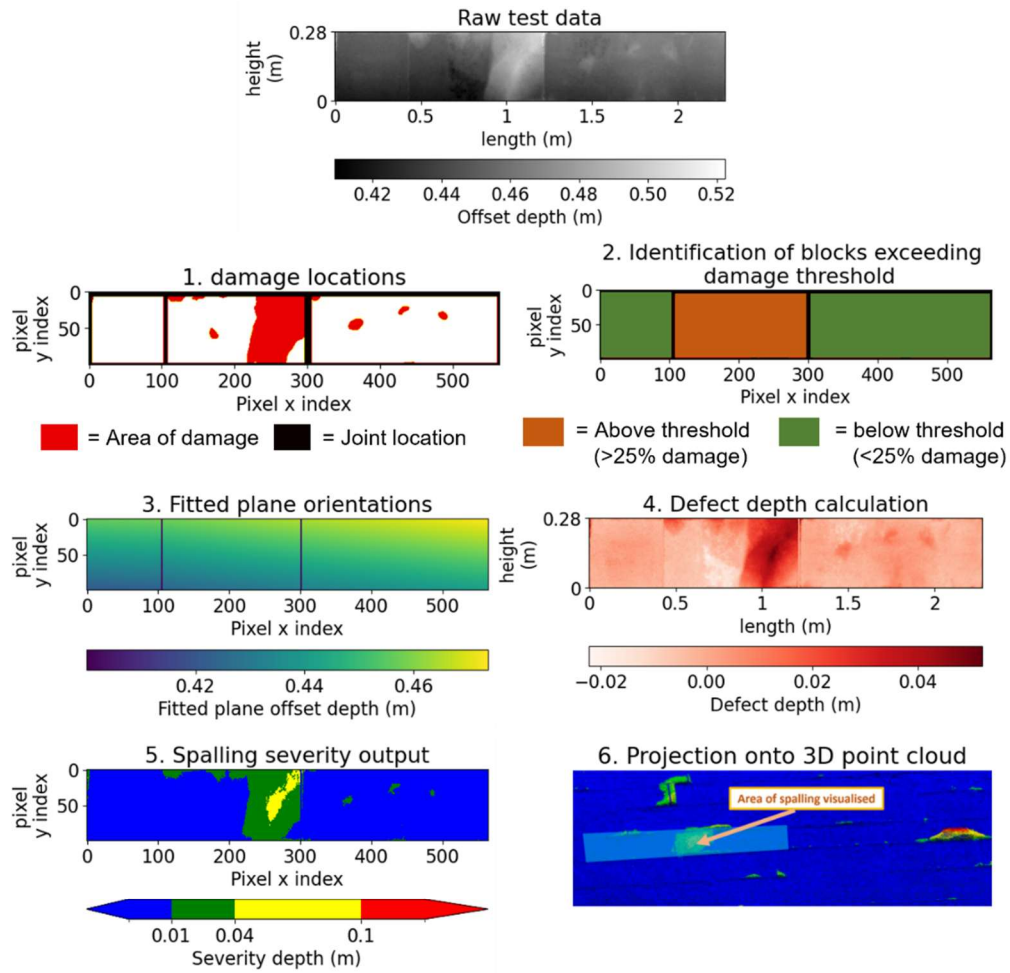
gradient set, the horizontal gradient of the block is set as a best fit between the two adjacent undamaged blocks.

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 Figure 5.13 (Network Rail, 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.

In order to demonstrate that the workflow's theory is effective across different lining situations, it was tested on samples of synthetic tunnel data that had been subject to different types of synthetic damages. The damage maps were created by thresholding any synthetic spalling or surface deposits added to the lining that were over 5mm magnitude. The masonry joint map was set as the masonry joint locations used to create the synthetic data. The workflow was evaluated for segmenting spalling damage with 10mm depth for a tunnel with:

a) Synthetic spalling and deformation
b) Synthetic spalling and surface deposits
c) Synthetic spalling, noise and surface roughness
d) Synthetic spalling, deformation, surface deposits, noise and roughness

The output of each of these segmentations is shown in Figure 5.14. The majority of spalling locations are correctly identified, with little change in performance when different damage types are applied. However, there is a slight under segmentation in some areas. This is due to the threshold selected for determining damages masked out by the damage map. If the threshold is set higher, then some areas of spalling will be undersegmented, as there will be a slight offset in the fitted masonry surface planes. But if the threshold is set lower, it will encompass small scale noise and substantially reduce the number of pixels available for fitting the surface plane, potentially leading to it being placed at an incorrect angle. A quantitative performance evaluation of the workflow is conducted in Chapter 9.



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

Figure 5.13: Top: Output of each stage of the plane fitting and severity thresholding algorithm for a three block section of Tunnel. Bottom: Network Rail masonry spalling severity thresholds (No Concern category non-standard but generally accepted).

5.5.5. Visualisation and Deformation analysis

After the plane fitting stage of the workflow, spalling severity maps are produced alongside maps of the unspalled face plane location of each block. Knowledge of each block's location enables visualisation of any lining deformation. From a single point cloud survey, it is not possible to conclusively determine the location and magnitude of tunnel lining deformations without knowledge of the as built geometry as a reference. However, the geometry along the tunnel can be assumed to be either straight or follow the profile of the track. As a result, relative radial lining deviations along the tunnel's length can be calculated by setting the cross section of the tunnel at one end as the reference radial geometry. The reference geometry is formed from the undamaged blockface output by taking the average of the first 20 pixels in each row (excluding rows containing a masonry joint) to generate a single column of depth map data. This column is then subtracted from every column in the complete undamaged blockface output depth map to show the radial deviation. Figure 5.15 shows how the radial deviation depth map created using this method compares with the ground truth radial deviations that were synthetically added to the data. It also shows the complete synthetic lining deviation map used to generate the synthetic point cloud and the difference between this and the identified radial deviations down the tunnel. Most of the error is caused by added circumferential deviations. Unless an as built cross-section is assumed, it is not possible to automatically separate circumferential deformation from the tunnel cross section, so it would need to be identified manually through visualising the tunnel lining point cloud.

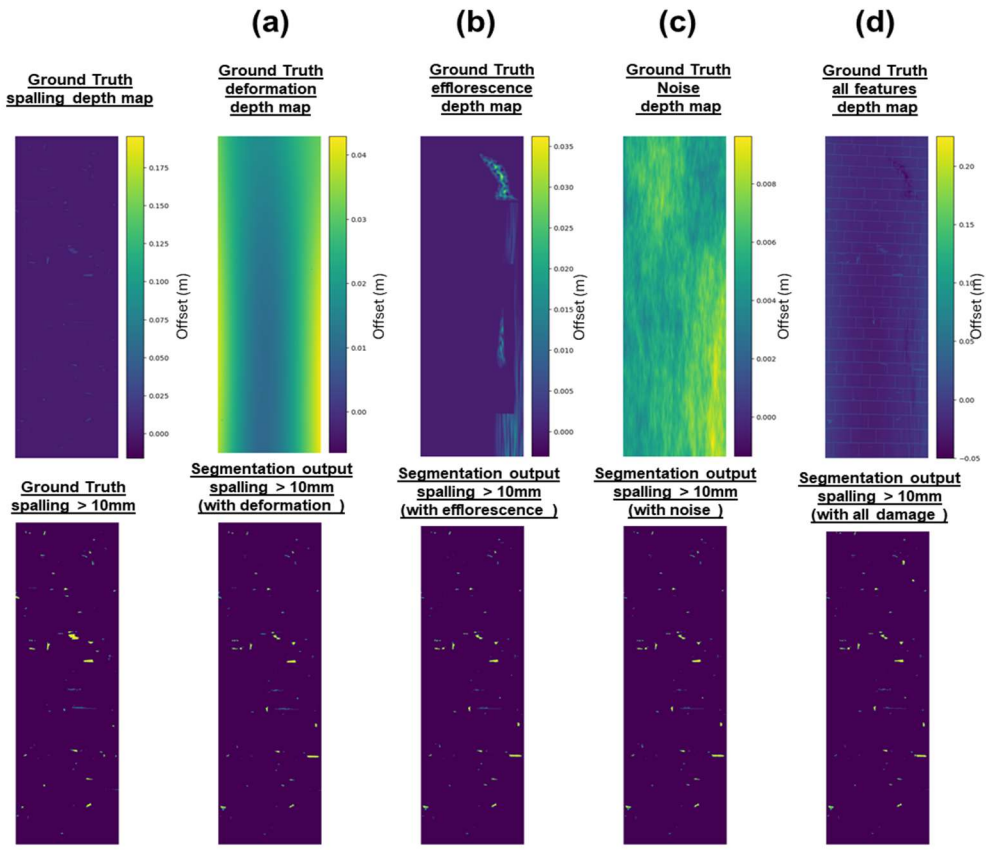


Figure 5.14: Visualisations to show the spalling segmentation output for spalling greater than 10mm deep on synthetic tunnel point cloud data with different types of damage added. a) Added synthetic spalling and deformation. b) Added Synthetic spalling and surface deposits. c) Added synthetic spalling, noise and surface roughness. d) Added synthetic spalling, deformation, surface deposits, noise and roughness.

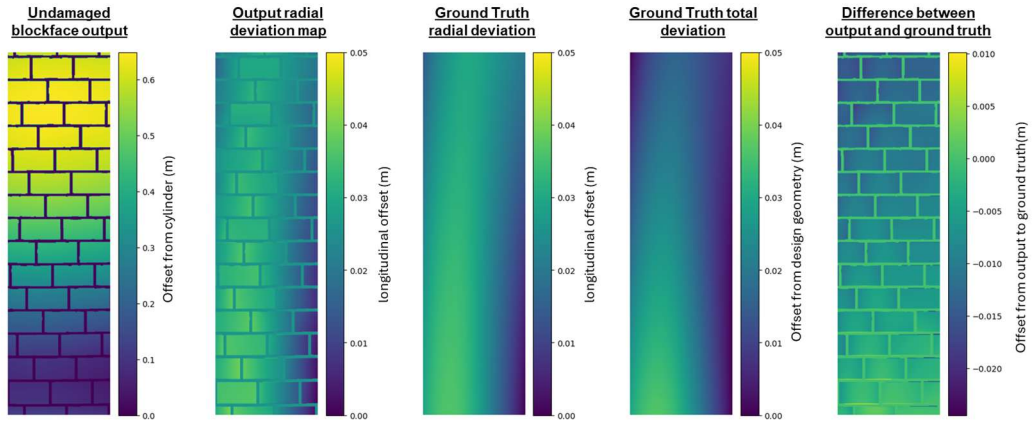


Figure 5.15: Demonstration of radial tunnel lining deformation visualisation.

5.6. Sensitivity analysis

It is expected that in real world applications, it will not be possible to semantically segment all lining damages and masonry joint locations for step 3 of the workflow with 100% accuracy. As a result, it is necessary to investigate how significantly the final spalling segmentation output is impacted by inaccurate damage and joint segmentations. Pixels in the ground truth damage and joint location ground truth masks were randomly flipped between the target and background class to model incorrectly generated masks. Masks were created with different proportions of pixels flipped and the accuracy of the new masks were calculated compared to the original. Each of the generated reduced accuracy masks were used within the workflow and the reduction in spalling severity accuracy was recorded and plotted in Figure 5.16. While high accuracies in both the damage and joint masks are required for the best spalling severity segmentation performance, the accuracy of the masonry joint mask is required to be much higher than that the damage mask needs to be before spalling segmentation accuracy is substantially impacted. At low damage segmentation accuracies, tunnels with larger synthetically generated masonry blocks, representing a stone lining, have their spalling segmentation accuracy reduced more significantly than for tunnels with smaller brick masonry. This is because false negative damage locations within a block will cause the undamaged face place to be erroneously fitted to spalled areas of the block, this will cause the surface of the whole block to be located incorrectly, leading to both false positive and false negative spalling locations within the block. These impact the accuracy over a larger proportion of the tunnel than for the smaller blocks within the brick lined tunnels.

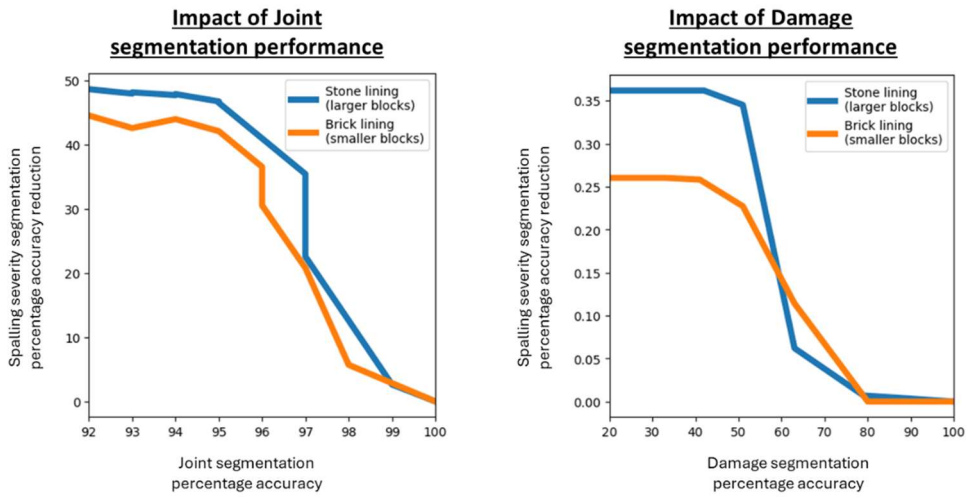


Figure 5.16: Percentage reduction in spalling severity segmentation accuracy for different accuracies of either the joint segmentation or damage segmentation maps used by the workflow. The graphs show the percentage accuracy reductions for 10mm depth spalling on synthetic lining data.

5.7. Required next steps

This chapter explained how spalling severity maps can be generated from 3D point cloud data of masonry tunnel linings. For the proposed workflow to be labor efficient in practice, the masonry joint and damage segmentation steps need to be automated. Masonry joint segmentation performance needs to be analyzed in particular detail since reduced joint segmentation accuracies are shown to cause a substantial breakdown in the workflow's performance. Reduced damage segmentation accuracy, on the other hand, only impacts spalling segmentation performance at lower accuracy levels and leads to smaller spalling severity segmentation accuracy reductions. There are multiple possible situations where the proposed workflow could be applied that the developed automated joint and damage segmentation methods should be designed to account for:

1. A method tuned on one section of tunnel before it is applied to the rest of the tunnel or to nearby tunnels with a similar design and condition. This requires manual labelling on each workflow application, alongside compute power and algorithm expertise to set up the selected algorithm on each application. If an algorithm can achieve acceptable performance using only a short section of tunnel for training/tuning, then it may still produce a more consistent and comprehensive joint and damage labelling than a human assessor would within a similar time. This method is most effective when there is a longer tunnel with a consistent design and condition. This situation is explored in Chapter 6.
2. A generalized masonry damage and joint segmentation method that has been pretrained/tuned on a selection of data so that it can be used "out of the box" without any additional training/tuning on a new

dataset. More substantial tunnel condition assessment time savings could be achieved with this method, although it is difficult to achieve given the wide variety of possible masonry types, geometries and damages that an algorithm may encounter. Chapter 7 analyses the generalizability of neural networks for masonry joint segmentation and assesses how consistent their performance is given different training and testing data splits.

3. It is important to be able to identify when a tunnel will be unsuitable for the method and segmentation performance will be too low for the workflow to be effective. This prevents the workflow from being used where it will produce more incorrect results, for example when the tunnel properties are out of the distribution used for training/tuning the algorithm. This will increase the trustworthiness of the method and the likelihood of adoption. Chapter 8 evaluates masonry joint semantic segmentation uncertainty quantification to assess when the method is applicable to a new tunnel.

The proposed workflow is assessed on real data in Chapter 9, where performance on real and synthetic data is compared and the practicality of the proposed workflow for condition assessments is evaluated.

6. Computer vision for masonry joint and damage semantic segmentation*

**This chapter includes parts of published works: (Smith et al., 2024, 2023; Smith and Paraskevopoulou, 2024)*

6.1. Purpose of chapter

The workflow proposed in Chapter 5 includes a step where the locations of masonry joints and masonry damages must be identified. Both of these tasks require an automated approach to avoid substantial manual labelling time. This chapter evaluates the feasibility of multiple computer vision algorithms to achieve this. In addition, by evaluating algorithms which operate directly on the 3D point cloud and those which require rasterised 2D images, the need for the workflows' tunnel unrolling and rasterization steps are assessed. The chapter focuses first on identifying the most effective approach for semantically segmenting masonry joints, before evaluating similar approaches for damage segmentation. While the masonry damage segmentation step is important for achieving more accurate damage severity values, accurate masonry block and joint segmentation is a prerequisite for the workflow to be feasible. Masonry joint segmentation also provides important tunnel documentation that can be used for scan to BIM and other workflows beyond that investigated within this thesis. Four categories of computer vision method were evaluated:

1. **Traditional Computer vision.**

   Filter and threshold based image analysis algorithms that do not use machine learning and operate on the rasterised point cloud. These

algorithms have few parameters which must be manually tuned during each application.

2. **Tree-based machine learning.**

Decision tree methods which classify individual points on the 3D point cloud using local geometric features. This supervised method requires pre-calculating point cloud features on the target dataset and training on example data.

3. **2D CNN**

Convolutional neural networks which classify pixels on 2D rasters of the unrolled tunnel point clouds. These supervised methods require training on example data. However, adapting existing pretrained models reduces the training time required.

4. **3D pointwise CNN**

These methods apply convolutional neural networks directly to the 3D point cloud. However, the field is less extensively studied than that of 2D CNNs and the methods are more computationally intensive.

This chapter assesses these methods in the context of applying the spalling severity segmentation workflow to a single tunnel containing a single masonry lining design. Where training is required, it would be conducted over a 10m section of tunnel, before the trained model could be applied to automate analysis of the remaining area of the same tunnel. 10m was chosen as a length that could reasonably be labelled within 4 hours and avoid substantial manual labelling that would erode the cost savings from automating analysis of the rest of the tunnel.

6.2. Datasets

Tunnel T1 was used as a case study masonry lined tunnel for this chapter. 80% of a 10m section of the tunnel was selected to use as the training set for each method if required, with the other 20% of this training section reserved as a validation set. A different 10m section was used as a testing set for evaluating the model performance. In order to effectively train and evaluate the models, ground truth masks needed to be manually created for each set. Binary masks of masonry joint locations were created using the 2D depth map rasters of each cloud created in step 2 of the tunnel assessment workflow. The QGIS software was used for labelling. (“QGIS_3.38.2,” 2025)

Exact masonry joint widths can be very difficult to determine from the point cloud data. Spalling at masonry block edges often creates a smooth pixel intensity gradient into the joint so it is challenging to delineate an exact boundary location. In some cases, the masonry joints are very narrow resulting in them not appearing in the image if less than the point density. Nevertheless, the semiregular nature of masonry patterns means it is possible to label a likely joint path when accurate pixelwise labelling is not possible. Due to the time-consuming and often inaccurate nature of manual pixelwise labelling, the ground truth joint maps were instead created as vectors of joint centrelines. In order to encourage each model to focus on ensuring the connectivity of each masonry joint, the joint vector maps were rasterised to constant 9-pixel width lines for training and testing. An example labelling of a section of tunnel is visualised in Figure 6.1. Semantic segmentation methods that operate on image data can use the rasterised depth map as their input and the joint map as the ground truth. For methods that use the point cloud directly, the labelled joint map was reprojected back onto the 3D point cloud data and saved as a scalar field.

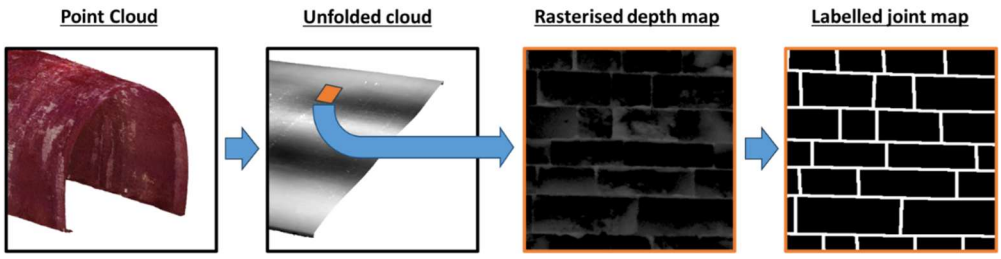


Figure 6.1: Visualisation of data labelling workflow

6.3. Traditional Methods

Non machine learning based algorithms may be used for semantic segmentation of image data. As part of the proposed spalling severity segmentation workflow, these methods can be applied to segment masonry joint locations from the rasterized depth map images. Research into computational enhancement and analysis of 2D images started to significantly expand in the 1970s, once compute power reached a level that became feasible to manipulate image data (Szeliski, 2022). Since then, many algorithms have been proposed for different computer vision tasks (Sezgin and Sankur, 2004).

6.3.1. Filter methods

For the task of binary semantic segmentation of images, individual pixels in the image need to be classified into one of two categories. In the majority of applications, this will be identifying a foreground object from the background. One approach is to assume that the edges of the target object are strongly defined from the background class. If the object's edge can accurately be identified, then it will form a dividing boundary between pixels belonging to the target object and the background. For the case of masonry joint segmentation, depending on the amount of mortar used, there is usually a distinguishable edge between either each masonry block or between block

and mortar areas that could be used to enclose each masonry block. Three popular edge detection methods were evaluated in this chapter. Prewitt, Sobel and Canny edge detection methods were applied to the testing data using the Scikit Image Python library (Van Der Walt et al., 2014).

Prewitt filters (Prewitt, 1970) are one of the most simplistic methods of edge detection. They involve convolving 3x3 vertical and horizontal edge detection kernels in turn with the image that highlight pixels at vertical and horizontal edges respectively. The Sobel filter is an alternative kernel to the Prewitt filter that is more robust to noise. Each kernel is shown in Figure 6.2.

An image kernel is a small square image which is placed over the input image, with the target pixel at its center. The kernel processes the local image structure by multiplying the value of each pixel in the kernel with the value of the pixel at the corresponding location in the input image beneath and then summing all of these values. The target pixel in the output image is set as this value, which for an edge detection kernel should have a larger magnitude if an edge is present and zero if there is no detected edge. This process is known as a convolution. The size of the kernel image determines the receptive field and the amount of local information used to determine if an edge is present. In order to find all of the edge locations, the kernel must be passed over the input image and this calculation conducted with the kernel centered on every pixel in the input image. The procedure is Illustrated for a vertically oriented Prewitt edge detection filter applied to an image with a vertical edge and one with a horizontal edge in Figure 6.3. The Prewitt kernels determine the existence of an edge by approximating the derivative of changes in pixel values in the horizontal or vertical directions.

A kernel cannot be applied to a pixel on the edge of an image without parts of the kernel overlapping beyond the bounds of the input image. To account

for this, the input image needs to be padded with additional pixels around the edges. Padding with zero intensity pixels often leads to unwanted edge effects. Figure 6.4 shows how when used with a vertical edge detection Prewitt filter, zero padding is misinterpreted as an edge, but replication padding is not. Replication padding copies the existing edge pixel intensities in the padding pixels. For the kernel convolution based methods used within this study, reflect padding was used. Reflect padding reverses the pixels at the edge of the original image in the padded area and is required when larger kernels are used to ensure that there is a smooth transition to the padded area while retaining the image texture at the edge.

The Canny edge detection method is a popular multistep kernel based edge detection method designed to detect only the most prominent edges in more complex scenes. It first uses a gaussian filter to smooth the image and reduce high frequency noise. It then uses Sobel filters to find the image gradients, before selecting a single line forming an edge by checking if there is a local maximum. Finally, the algorithm selects only the most prominent edges by using hysteresis thresholding to select the relevant local maxima (Canny, 1986).

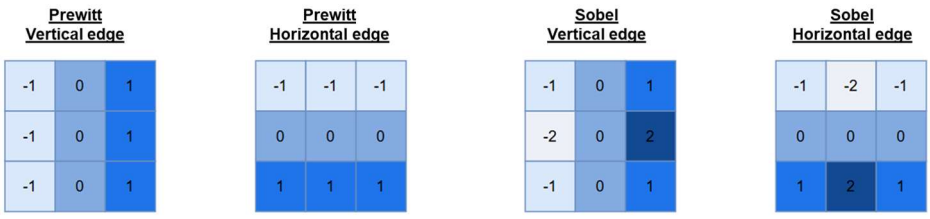


Figure 6.2: Kernels used for identifying vertical and horizontal edges using the Prewit and Sobel methods.

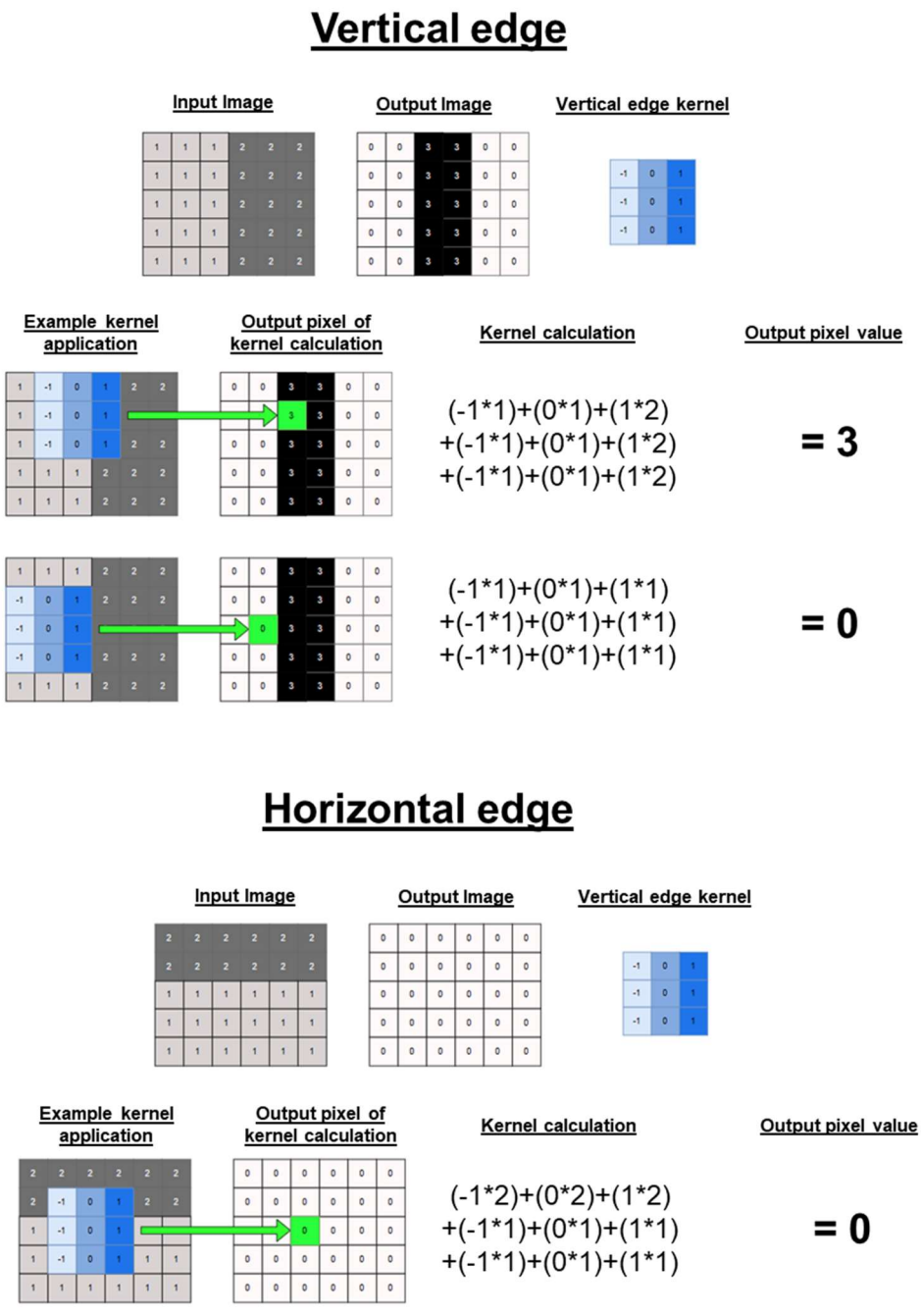


Figure 6.3: Visualisation of convolution of vertical edge detection Prewit kernel with an image containing a vertical edge and one with a horizontal edge.

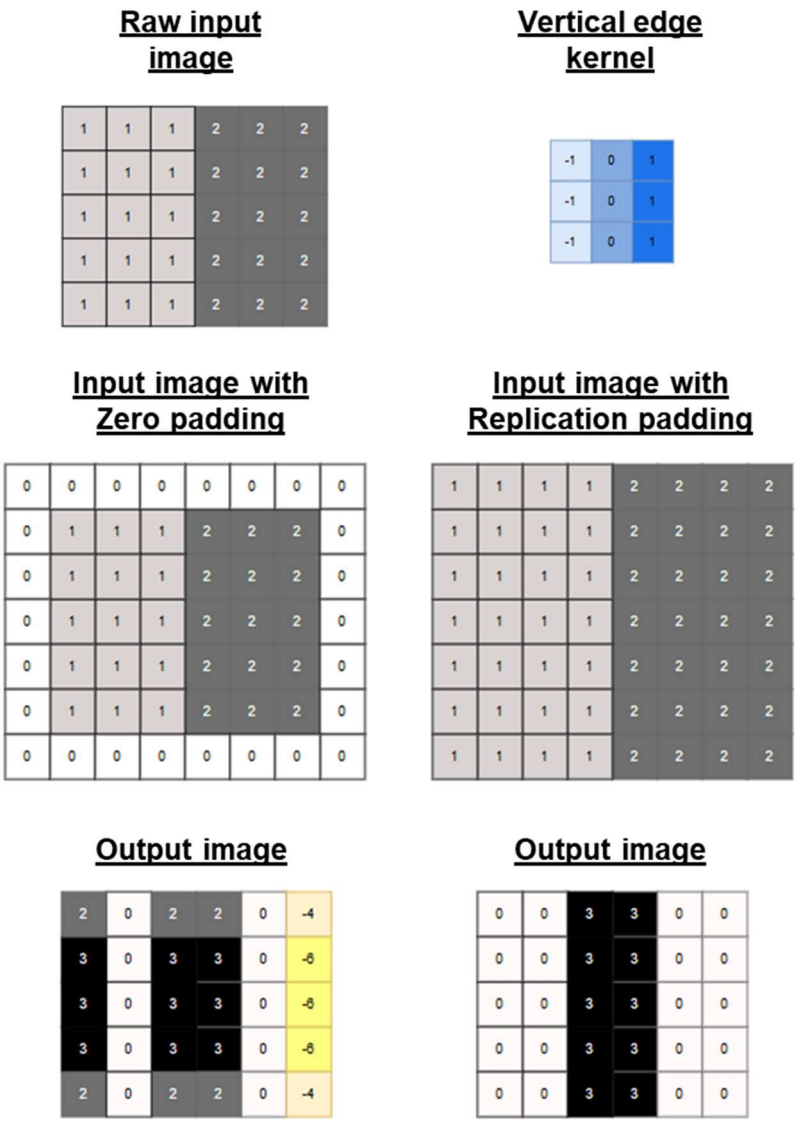


Figure 6.4: Visualisation of effect of using zero and replication padding with a vertical Prewit kernel.

6.3.2. Thresholding

An alternative approach to semantic segmentation of image data that does not use machine learning applies a threshold to pixel intensities and assumes that an object is either brighter or darker than the background. A threshold pixel intensity is set and pixels with a greater intensity are assigned to one class and those with a lower intensity to the other. For semantic

segmentation of masonry blocks and joints from depth map images, where masonry and mortar are in a good condition, the masonry blocks should protrude into the tunnel beyond the joint area, leading to differing intensities within the depth map. The most straight forward approach involves setting the threshold as the mean of all pixel intensities. However, this does not take into account that there may be a class imbalance with more pixels belonging to one class over the other. Alternative algorithms have been created that automate finding the optimal threshold.

Many algorithms use the histogram of image intensities to determine the optimum threshold. If the difference between intensities of pixels in each class is well defined, then there should be peaks in the intensity histogram at the typical intensity of pixels in that class, and a trough in frequency at intensities between the classes. An optimum threshold would be in the frequency trough. However, the situation is often complicated by a noisy background, leading to multiple peaks and troughs in the histogram. Furthermore, some pixels in the target object may have the same intensity as those in the background, rendering a complete segmentation impossible with a single threshold. For masonry joint segmentation specifically, a thresholding method will need to be robust to the existence of deviations in the surface profile caused by damages and lining curvature.

In order to simplify the thresholding task, images can be enhanced before thresholding is applied. These methods seek to maximise the local differences between the target object intensity and the background class and discard global intensity variations. If there are contrast variations across an image that are unrelated to the target segmentation task, then CLAHE (Contrast Limited Adaptive Histogram Equalisation) can be applied to match the intensity histograms between local image patches and ensure that the

distribution of intensities is similar in different parts of the input image. An alternative is to simply apply thresholding locally. Images are split into patches and then a thresholding method is applied to each patch in turn. A final method to remove global image variations and enhance local features is known as gaussian differencing. A large scale gaussian blur is applied to the image and the blurred image (which now has had local features removed) is subtracted from the original image to leave only local features. The size of the gaussian blur kernel applied is proportional to the scale of features removed. For a sample of T1 data, the output of CLAHE and gaussian differencing is shown in Figure 6.5.

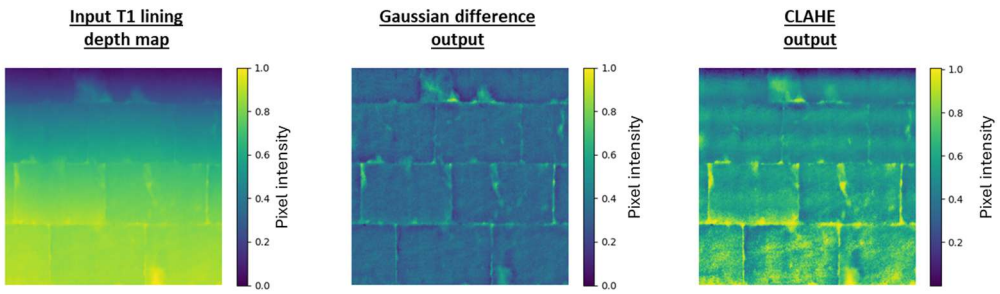


Figure 6.5: Visualisation of different image enhancement techniques applied to a section of tunnel T1 lining depth map.

This chapter analyses the following thresholding methods:

1. **Minimum Thresholding.** The Minimum thresholding method proposed by (Prewitt and Mendelsohn, 1966) assumes that there are only two objects in the input image and applies smoothing algorithms to the intensity histogram until there are only two maxima. The segmentation threshold intensity is then set as the intensity with the minimum frequency between the two maxima in the smoothed histogram.

2. **Mean Thresholding.** The threshold is set as the mean of all pixel values
3. **Yen.** The Yen thresholding algorithm (Yen et al., 1995) attempts to find a threshold that minimizes the information loss from the input image to the thresholded one by using a maximum correlation cost function.
4. **Triangle.** Designed for unimodal histograms with long tails, Triangle thresholding attempts to split the tail area from the peak region by drawing a line from the top of the highest peak in the intensity histogram to the end of the tail. The threshold is set as the point on the tail side of the histogram with the greatest perpendicular distance to the line.
5. **Isodata.** This histogram-based thresholding method, known as Ridler-Calvard (Ridler and Calvard, 1978) applies the threshold midway between the mean intensity of the classes either side of the threshold.
6. **Otsu.** The Otsu method (Otsu, 1979) conducts an exhaustive search for a threshold that maximises the intraclass variance. The histogram of intensities of each class either side of the threshold must be generated and the variance of each distribution calculated in order to find the intraclass variance. Although Otsu thresholding is the most popular thresholding method, since it is exhaustive, it can be expensive to compute for higher precision images.
7. **Li.** Li thresholding (Li and Tam, 1998) uses an iterative method to find the optimum threshold. It minimizes the cross entropy between the background and foreground mean values to find the threshold that best separates the distributions.

6.3.3. Decision Trees

Decision trees are a popular supervised machine learning method that select a classification based on the values of multiple 1D input parameters (de Ville, 2013). For the task of masonry joint segmentation, decision trees can be applied to the input 3D point cloud directly without rasterising each point. Points in the 3D point cloud can be assigned features which describe aspects of the geometry. The decision tree can then be applied on a pointwise basis to classify each point.

A decision tree asks a series of questions about the data by thresholding to decide on the target class. The specific question asked is dependent on the result of the previous question. The tree of thresholding questions is typically created using a greedy algorithm that designs each node as the most effective classifier at the current stage of tree design without considering the performance of subsequently generated nodes. Many other algorithms for generating decision trees have also been proposed (de Ville, 2013). As the nature of each question is a simple threshold, the reasoning behind the final decision is explainable. Decision tree methods have been made more powerful at the expense of explainability through the advent of the Random Forest method (Breiman, 2001) and Gradient boosting using the XGBoost algorithm (Chen and Guestrin, 2016). These train multiple decision trees to improve test data performance.

Decision trees are prone to overfitting, especially when a few parameters in the training data appear to have an oversized importance in generating the decision outcome. These variables will cause the others to be ignored during the training procedure. The Random Forest method seeks to avoid this problem and regularize the model by creating multiple trees, but only using a subset of training features for each tree. The final decision is the average

output of all the generated trees; this also makes the method less sensitive to noise.

XGboost (eXtreme Gradient Boosting) has become the “go to” decision tree based method due to its increased speed and classification performance compared to other tree based methods (Nielsen, 2016). Boosting iteratively creates multiple decision trees in succession. After the creation of each tree, the residual error of the tree is calculated, and the next tree is generated to account for the error. XGBoost is a boosting algorithm that is designed for high performance on larger datasets by optimizing memory usage and enabling parallel processing and GPU acceleration. It also incorporates regularization and pruning methods to prevent overly complex trees from forming, reducing overfitting.

For this study, due to its popularity and improved computational performance (Mienye et al., 2019), XGBoost was chosen as the representative decision tree based method. Local geometric features were created for each point in turn using the methods first proposed by (Hackel et al., 2016a). These features have previously been used with decision tree based point clouds to classify street furniture (Hackel et al., 2016b) and parts of historic building facades (Grilli et al., 2019). In addition, (Grilli et al., 2019) showed Random Forest and One-versus-One (OvO) classifier methods applied using these features outperformed more modern deep learning based methods for historic building part segmentation. This performance was attributed to the difficulty in applying neural networks to unordered point cloud data. However, they trained and tested each technique on different parts of the same structure, so it is uncertain how well the method generalises and whether the classifiers would be effective when applied to other similar structures.

The features selected use relationships between points in a subcloud of points within a specified radius around the target point to characterise the local geometry. The features are relationships between the first, second and third eigenvalues of the subcloud. The point cloud's eigenvalues represent the spread of points in different orthogonal directions, with the first calculated in the direction with the most variance, the second in an orthogonal direction with the second highest variance and the third in the final orthogonal direction with the third highest variance. Calculation of these features involves first generating the point cloud K-d tree in order to organise the cloud and enable the calculation of each point's nearest neighbours. 0.05m and 0.01m were selected as the nearest neighbour radii., as the average joint width within the dataset is 0.01m. These distances should therefore characterize both features solely within a joint and those that represent how a joint appears within the context of the broader point neighbourhood. In addition to the spatial features developed by Hackel et al. (2016), surface roughness was used. This is defined as the offset of the target point from a 2D plane fit to the remaining points within the neighbourhood. Where $\lambda_1, \lambda_2, \lambda_3$ are the first, second and third eigenvalues respectively of the points in the local neighbourhood, Table 6.1 shows each eigenfeature used for input into XGBoost.

In this chapter, The XGBoost model was then trained with a learning rate of 0.1 for 599 epochs. After the selected features were calculated for the input point cloud before training, the number of block and joint points was balanced to ensure equal importance is given to each class. The optimum number of epochs was selected after the validation loss did not decrease below its minimum achieved within the trailing 20 epochs.

Table 6.1: Eigenfeatures used within this study as input features to the XGBoost classifier.

Feature	Calculation
Sum	$\lambda_1 + \lambda_2 + \lambda_3$
Omnivariance	$(\lambda_1 \cdot \lambda_2 \cdot \lambda_3)^{\frac{1}{3}}$
Eigenentropy	$-\sum_{i=1}^{3} \lambda_i \cdot \ln(\lambda_i)$
Anisotropy	$(\lambda_1 - \lambda_3)/\lambda_1$
Planarity	$(\lambda_2 - \lambda_3)/\lambda_1$
Linearity	$(\lambda_1 - \lambda_2)/\lambda_1$
Surface Variation	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$
Sphericity	λ_3/λ_1
Verticality	$1 - \lvert [0\ 0\ 1], e_3 \rvert$
1st Eigenvalue	λ_1
2nd Eigenvalue	λ_2
3rd Eigenvalue	λ_3

6.4. Deep learning-based methods

Deep learning techniques have proven to be the most effective methods for semantic segmentation across many tasks and disciplines (Asgari Taghanaki et al., 2021; Long et al., 2015; Luo et al., 2024; Xie et al., 2019; Yuan et al., 2021). The large number of learnable parameters and the ability of Convolutional Neural Networks (CNNs) to operate efficiently on image data allows them to be effectively trained for most segmentation tasks (Voulodimos et al., 2018). As a result, methods involving CNNs currently

form the state of the art in semantic segmentation technology (Mo et al., 2022).

There are four main approaches for applying defect detection and segmentation neural networks to 3D data: point based, voxel, Multiview and single surface projection. A comparison of the key methods is shown in Table 6.2, 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).. The majority of studies considering the analysis of surface defects use a Multiview approach. Reducing the complexity of the problem to 2D makes training models easier and takes advantage of the more advanced state of 2D computer vision research.

The first, and most challenging, approach that this chapter evaluates is deep learning algorithms applied directly on the 3D data points. As point cloud data is unstructured, relationships between neighbouring points are difficult to determine and computationally expensive to characterise. PointNet (Charles R Qi et al., 2017) formed operates directly on point cloud data and has been shown to be effective for shape segmentation. Since then, further methods have been proposed that are discussed in more detail in section 6.4.2.

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

Multiview projection involves projecting the point cloud onto multiple planes of differing orientations and rasterising the distance of each point from the plane on 2D images. 2D CNN –based neural networks are then used to characterise 3D features by integrating knowledge of perspective of the different images into standard 2D image analysis techniques. The field is not well developed, so this method has not been used extensively for condition assessment in any discipline. For a tunnel lining that can be unrolled, there is limited benefit to applying a Multiview method over a single surface projection based one, so this chapter considers only a single surface projection-based 2D approach.

An alternate method is to rasterise a point cloud into a 3D data grid using voxelisation. While this has been used for the creation of tunnel digital twins (Wu et al., 2022), it has not been used for machine learning on tunnel data, due to the computational complexity of training a model on a sparse 3D data grid containing only a tunnel lining surface. Much research has been undertaken to detect objects and segment scenes from Mobile laser scanners with Voxel based methods for application to autonomous cars. (Zhou and Tuzel, 2017) for example created VoxelNet, a modern Voxel method to detect pedestrians, cars and bikes. (Ji et al., 2022) successfully segmented water seepage and installations from a concrete tunnel lining. They developed a 3D encoder-decoder network that operated on the Voxelised point cloud. This was shown to outperform VoxNet and also PointNet and PointNet++ which operate directly on the individual points.

(Vora et al., 2020) created the PointPainting sequential fusion method to take advantage of situations where both photographic and lidar data are available. PointPainting projects the output of a 2D image semantic segmentation network onto a 3D point cloud obtained by lidar, before applying a 3D segmentation method. This produces superior results to 3D only object detection techniques. Nevertheless, Voxel based methods are not considered in this study due to their inefficiency when applied to segment parts of the single surface that forms a tunnel's lining.

Table 6.2: 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), KPConv (Thomas et al., 2019)	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. Conventional computer vision techniques can be adapted to work in three dimensions and applied to this 3D image.	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-R-CNN (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).

Method	description	Major use cases	Relevant architectures	Advantages and Disadvantages
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 (Su et al., 2015), MHBN (Yang and Wang, 2019), View-GCN (Wei et al., 2020)	Integration of standard 2D CNNs enables good segmentation of local features in addition to global 3D shapes. The method does not fully exploit the 3D nature of a point cloud and does not work well when surfaces are irregularly shaped. As a result, some points may be lost through occlusion and success is dependent on view selection. The field is also relatively undeveloped.
Single surface projection	The point cloud is projected onto a surface and unwrapped, then rasterised into pixels on a 2D image. These may simply take the RGB or intensity data of the cloud or may be a depth map indicating the distance of the surface of each point from the viewpoint. 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 lining's using a cylindrical projection (Feng et al., 2023; Huang et al., 2020b; Zhou et al., 2021).	CNNs (Simonyan and Zisserman, 2014a), U-Net (Ronneberger et al., 2015) , UNet++ (Z. Zhou et al., 2018), SegFormer (Xie et al., 2021)	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).

6.4.1. 2D Deep learning

2D encoder-decoder style CNNs have become the preferred method for supervised semantic segmentation of image data and single surface projection enables them to be used for point cloud semantic segmentation. Before the advent of the U-Net, semantic segmentation was often achieved using only encoder style CNNs with a pixelwise or sliding window strategy. An outline of different approaches to semantic segmentation is shown in Table 6.3.

The U-Net is now one of the most popular architectures for semantic segmentation, as it was the first to demonstrate excellent results with transfer learning and only limited domain specific training (Ronneberger et al., 2015). Shown in Figure 6.6, it follows a standard encoder-decoder design, with added skip connections between the encoder and decoder. An encoder has a similar design to convolutional neural networks used for classification. The decoder combines the feature maps from the encoder's output and up-samples them to increasingly higher resolutions to attempt to recreate the image, but with each pixel labelled with its classification. At each level of the encoder, the output feature maps are fed both to the next level of the encoder and to the skip connections. At equivalent levels of the decoder, the output of the decoder level below is upsampled and concatenated with the skip connection value. This enables 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.

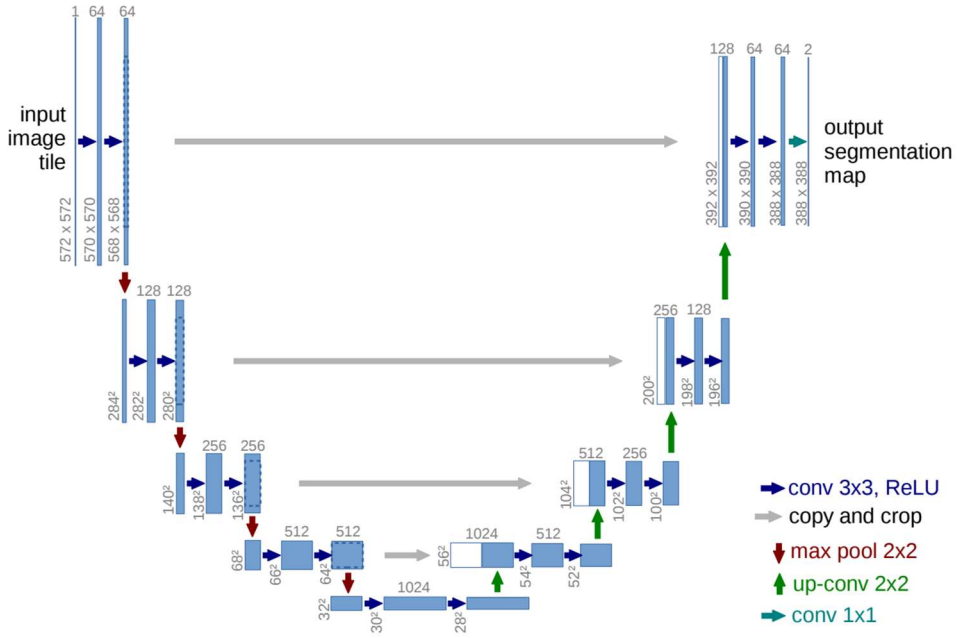


Figure 6.6: U-Net architecture, from Ronneberger et al. (2015)

There are often many common features between semantic segmentation classes. For example, segmenting brick locations in brick wall images will be similar to segmenting stone locations on images of stone walls. As a result, transfer learning can be applied effectively to U-Nets, reducing the amount of subject specific training data required. This enables a network to be trained thoroughly on a large and general set of images and classes not directly linked to the target problem, for example the freely accessible ImageNet data set initially developed by (Deng et al., 2009). Then, the network can be briefly trained on the required problem, reusing the feature maps created in general training, but adapting to the required output classes.

For this chapter, the U-Net architecture was applied to segment masonry joints from the tunnel lining depth map images using ResNet34 as the encoder backbone. The ResNet (He et al., 2016) introduces an identity shortcut connection to the classic VGG encoder design introduced by (Simonyan and Zisserman, 2014b). This allows the model to skip one or

more layers and reduces the issue of vanishing gradients. In general, a deeper network can contain more parameters and feature scales so that more complex features about an image can be learnt. However, within a standard encoder design, performance reduces with very deep networks. This is because the gradients of the neurons within each layer with respect to the final classification become vanishingly small, so it is difficult to train them to converge and their information is lost. The ResNet makes the network learn residual functions instead of unreferenced functions. These are easier to optimise and enable the model to focus on information in the most important layers.

Table 6.3: Machine learning based semantic segmentation techniques

Image segmentation Method	Description	Advantages	Disadvantages
Pixelwise classifier	Applying a model to classify each pixel in turn based on CNN generated or handcrafted image features from neighbouring pixels.	Simpler algorithms can be used that require less compute to train.	Long pixelwise inference time. Output can be noisy.
Patchwise classifier	Splitting an image into patches, then applying a CNN classifier, usually an encoder-based design, to classify each patch.	Easy to train, clearer output on noisy data.	Localisation of damage is limited to the size of the patch.
Sliding window classifier	Similar to Patchwise but using many overlapping patches and combining the result of each patch that overlaps with each pixel to produce a pixelwise segmentation.	Achieves semantic segmentation with simpler neural networks.	Longer inference time. Segmentation boundaries are less accurate.
Encoder-Decoder style segmentation network	Neural network conducts pixelwise segmentation on image patches. Most designs use an encoder to create a feature description of the image at different scales, then use a decoder to create a segmentation map from the description.	State of the art accuracy, good boundary definition and transfer learning performance with less training data.	Requires more significant tuning and compute power to achieve the best performance.

Before a neural network can be trained, its optimiser and loss function need to be defined. A loss function is a differentiable equation used to compare the output of a neural network with the ground truth (ideal target output). Through backpropagation, the network's optimiser will adjust the network to minimise the value of the loss function. As a result, setting an appropriate loss function is key to ensuring that a neural network learns appropriately to minimise the differences between the network output and ground truth and focuses on the features that impact the quality metrics relevant to the target problem. For the task of semantic segmentation, most studies use loss functions set as the negative of standard performance metrics that aim to minimise the pixel differences.

Pixelwise accuracy is the simplest performance measure and in the case of semantic segmentation is equal to the proportion of total pixels that are correctly segmented. However, it does not account for class imbalance. IoU (Intersection over Union) is the most popular metric for semantic segmentation, as it provides a holistic overview of performance of the target class. This means that it is not affected by class imbalance. A diagram is shown in Figure 6.7. Where TP is the number of true positive pixels (Correctly identified joints), TN is the number of true negative pixels (correctly identified blocks), FP is the number of false positive pixels and FN is the number of false negative pixels, IoU is calculated as follows:

$$IOU = \frac{Intersection}{Union} = \frac{TP}{TP + FN + FP}$$

An IoU of 0.5 is often considered as a target level of segmentation quality for many tasks (Shah, 2023), although the exact threshold for acceptable performance is dependant on the nature of the downstream task. No direct IoU performance comparisons are made in this thesis between the results

generated and those presented in other studies because performance is heavily dependant on the exact datasets used for training and testing in addition to the quality of the ground truth masks used.

Ideally, a loss function is tailored to a target problem and optimises the same segmentation quality metric used for analysis of the test dataset. Using metric-centred loss functions has been proven to lead to improvements in the equivalent quality metric on test datasets (Eelbode et al., 2020). As a result, soft Dice loss has become the most popular loss function for semantic segmentation. Soft Dice is calculated by adding smoothing terms to the Dice metric calculation and has been shown to be effective for semantic segmentation tasks with large class imbalances (Sudre et al., 2017). Dice score uses a similar calculation to IoU that less strongly penalises false positive and negative segmentation. A similar soft IoU (Jaccard loss) has also been created; however it has not been shown to have any benefits over soft Dice (Bertels et al., 2019).

While parameter gradients can be calculated through backpropagation, optimisers provide the method for iterating towards a state that minimises the loss function. Three different optimisers were trialled, outlined in Table 6.4.

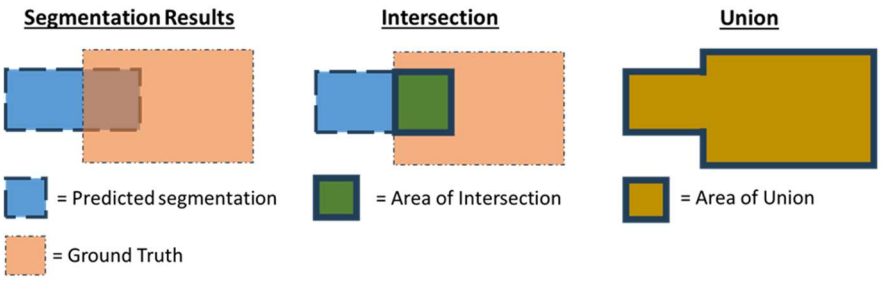


Figure 6.7: Visualisation of IoU calculation

Table 6.4: Optimisers trialled during hyperparameter tuning

Optimiser	date	Description
Stochastic Gradient Descent (SGE)	1950s	SGE calculates the gradient of the loss function with respect to the model parameters and updates them in the direction of the negative gradient by the amount specified in the learning rate.
ADAM (Kingma and Ba, 2014)	2015	The Adam optimiser adds adaptive momentum to the SGE method. This updates the learning rate iteratively based on the second derivative of the gradient. This results in faster convergence rates for the same level of accuracy.
ADAMW (Loshchilov and Hutter, 2017)	2017	AdamW adds controlled weight decay to ADAM. This fixes an issue with ADAM where weight decay can have varying effects, as it does not correctly scale with the change in learning rate.

When training a neural network there are multiple hyper parameters that need to be tuned to maximize a network's performance. In this section, a search was conducted to find the optimum hyperparameter combination to use with the U-Net and ResNet34 architecture. Figure 6.8 shows the overall procedure used for generating the proposed CNN model. Table 6.5 outlines the methods and hyperparameters used for training the 2D CNN. For some parameters where it is likely that the optimum value will stay constant across other hyperparameter combinations, trial and error was used to find the optimum setting. Other hyperparameters were selected using a grid search. This is a resource intensive procedure where every possible hyperparameter combination is trialed. The purpose of each parameter is explained in Table 6.5. The hyperparameter combination that generated a trained network with the highest IoU on the test data was selected as the optimum combination and the chosen values are highlighted in bold in Table 6.5.

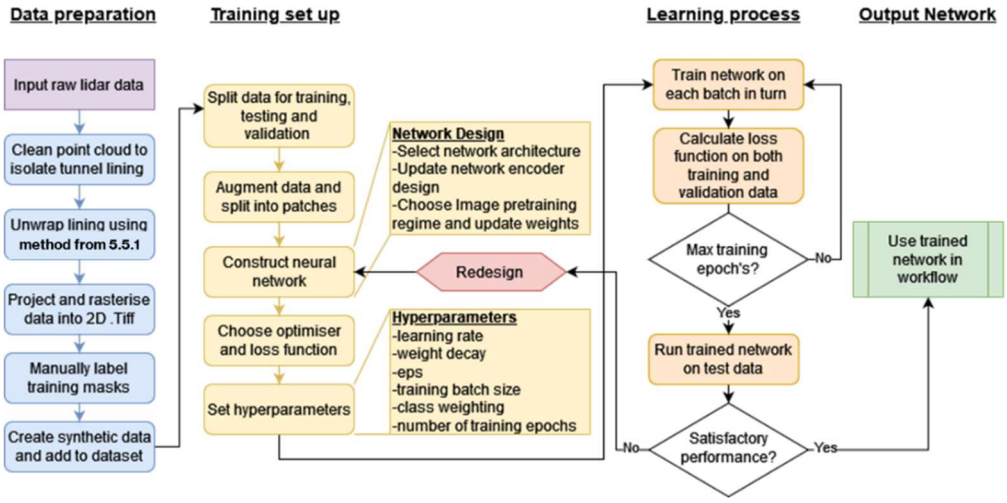


Figure 6.8: CNN tuning and training procedure

Table 6.5: Methods and hyperparameters evaluated. Selected combination highlighted in bold.

Parameter	Description	Values	Search type
Architecture	Overall design of neural network including decoder and connections between encoder and decoder.	**U-Net**	Fixed
Encoder	Backbone of the network, specifying the number of layers and specific design of the encoder.	**ResNet34**	Fixed
Optimiser	The optimiser forms the algorithm used during training to adjust the neural network's parameters through backpropagation to minimise the loss function.	SGE, Adam, **AdamW**	Trial and error
Initial learning rate	This determines the step size between training iterations. There is a trade off between slow learning and overshooting the optimum minima.	0.00001**,** **0.0001,** 0.001, 0.01	Grid
Image dimension	The vertical and horizontal dimensions of the input images and ground truth masks in number of pixels. Larger images enable larger scale image features to be better characterised, but there will be less training data.	512, **384**, 256, 128	Grid
Pretrain weights	Whether the neural network has been previously trained on a different dataset before resuming training on the images used in this study.	Random, **ImageNet**	Grid
Eps	An offset term added to the denominator of calculations to improve numerical stability and neural network convergence.	0, 0.0001, **0.0002**, 0.0004, 0.0008	Grid
Batch size	The number of images passed through the network simultaneously during training. Larger numbers give faster training but are limited by the device's VRAM.	**8**	Trial and error
Loss function	The loss function calculates a performance metric that the optimiser seeks to minimise through training.	**Soft Dice**	Fixed
Max training epochs	The maximum number of epochs that the method is trained for. All methods converged within 1200 epochs.	**1200**	Trial and error

6.4.2. 3D Deep learning

Neural networks that operate directly on point clouds bring multiple benefits over applying 2D CNNs with a single surface projection approach. They offer the potential to operate with any tunnel geometry without pre-processing and avoid any data loss associated with point pooling during rasterisation. However, they face the issues when dealing with point clouds as outlined in section 5.1.

PointNet (Charles R Qi et al., 2017) was the first deep learning method effectively applied to point cloud semantic segmentation. It first applies a spatial transformer network to align the point cloud in feature space so that similar features are in similar locations. This enables it to robustly work with objects in different orientations and spatial locations. It passes each point individually through a multi layer perceptron to extract local features and then uses a max pooling layer as a symmetric function to create a global feature vector from the cloud so that it can gain invariance to possible permutations of point order. PointNet achieves good performance for classifying defined objects, but is not good for segmentation of multiple objects or object parts in a scene, as it does not scale well to scenes with many points without losing detail and does not capture local relative information between points.

PointNet++ (Charles R. Qi et al., 2017) was a subsequent development of the architecture aiming to fix the key issues with PointNet. PointNet++ applies PointNet recursively to capture features at different scales. Its employs multi scale grouping of points within a region to provide better contextual information about each point. Adaptive neighborhood sizes enable characterization of clouds with varying point densities.

These architectures have been successfully applied, for example, by Ji, Chew, Xue and Zhang (Ji et al., 2022) to segment objects such as railway track, service pipes and tunnel lining locations within a point cloud. It has not been tested extensively for defect segmentation, although (Bolourian et al., 2023) used a modified PointNet++ to operate on point normal vectors, colour and location data. They trained the model to segment concrete cracking and spalling. (Pierdicca et al., 2020) applied deep learning methods to segment architectural components in historic structures. They compared PCNN, PointNet, PointNet++, Dynamic Graph Convolutional Neural Network (DGCNN) with a modified DGCNN. The modified DGCNN achieved the best F1-Score. Nevertheless, PointNet++ (Charles R. Qi et al., 2017) still doesn't consider local spatial relationships so is ill suited for identifying the minor surface deviations distinguishing masonry blocks from joints on a tunnel lining.

KPConv (Thomas et al., 2019) was later designed to improve the detail of the segmentation by better characterizing local point relationships. Inspired by 2D convolutions, it introduces 3D Kernel point convolutions that can operate with a varying number of points directly on the point cloud. The convolutions operate on nearest neighbour points with the contribution of each point weighted by Euclidean distance from the convolution centre. These convolutions can also be made deformable, which allows them to adapt their shape to different local geometries. Using the deformable kernel version achieves superior performances when target objects have a similar shape, but risks overfitting to shapes present in the training data. The method requires first structuring the point cloud by calculating the KD tree enabling point nearest neighbour calculations.

This study uses KPConv (Thomas et al., 2019) as the point based neural network method for masonry joint segmentation. It was selected because it is better at segmenting parts of connected objects compared to PointNet based models which better characterize whole objects within a scene (Soilán et al., 2020). The 3D convolutions should enable it to characterize the localised geometric deviations in the tunnel lining forming masonry joints.

The KPConv network was examined with both a rigid and flexible kernel. This enabled us to study the impact of kernel deformations on the results. To examine the how well KPConv can generalize features to different plane orientations with limited but representative training data, the method was assessed with both the raw input tunnel point cloud and the unrolled point cloud created for the 2D image based methods. Random rotations around the Z axis were applied to the training data to account for any possible tunnel orientation. The network was trained with a learning rate of 0.001. The maximum radius of the convolution was selected iteratively as 0.15m. Due to the substantial computational cost for training KPConv, ARC4 was used for training. Training was capped at a maximum of 48 hours.

6.5. Results

6.5.1. Traditional Computer vision results

Canny edge detection was applied directly to the test data with different gaussian filtering widths (σ). σ*4 defines by the width in pixels of the applied gaussian filter. The gaussian width determines the level of smoothing applied and how much noise is filtered from the output. Outputs on sections of stone and brick masonry within the testing dataset are shown in Figure 6.9. Table 6.6 details the performance by gaussian width and shows that a greater width initially primarily reduces the number of false positive segmentations, before

reducing both true positive and false positive segmentations at higher values. This leads to oversegmentation at lower values and undersegmentation at higher values. The best performing filter uses a gaussian width of 3.

8 thresholding methods were also trialled. Example outputs are shown in Figure 6.10 and performance is outlined in Table 6.7. The Otsu, Li and Mean thresholding methods achieve much higher IoU scores than the other methods, although the number of false positive pixels is higher than both the other thresholding methods and Canny edge detection. Given the roughness of the masonry surface and the existence of spalled areas, it is not possible to apply a single threshold to segment joints without also segmenting other recessed areas of masonry. Although Otsu thresholding achieves the highest IoU amongst these traditional computer vision methods, observing the segmentation maps shows that the Canny method achieves better joint closure so would potentially be a better choice for use in the spalling severity segmentation workflow.

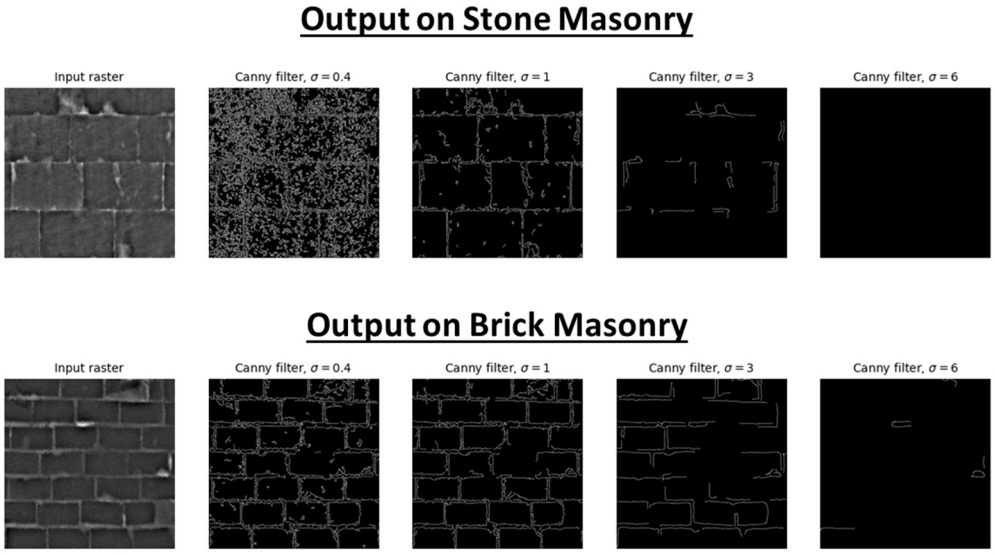


Figure 6.9: Canny edge detection applied to a section of brick and stone masonry in tunnel T1 with different sigma values (gaussian filtering width)

Table 6.6: Performance of Canny edge detection with different gaussian widths on the test data. The best performing width by IoU is shown in bold.

σ	IoU	precision	recall	accuracy
0.5	0.004	0.176	0.004	0.839
1	0.039	0.168	0.050	0.810
2	0.179	0.322	0.288	0.790
3	**0.185**	**0.387**	**0.261**	**0.817**
4	0.165	0.413	0.215	0.827
5	0.136	0.422	0.168	0.831
6	0.105	0.424	0.122	0.834

Table 6.7: Performance of different threshold selection methods, with the best performing method by IoU highlighted in bold.

Method	IoU	precision	recall	accuracy
Otsu	**0.198**	**0.273**	**0.419**	**0.730**
Minimum	0.000	0.000	0.000	0.841
Triangle	0.030	0.143	0.036	0.812
Yen	0.001	0.817	0.001	0.841
Li	0.195	0.255	0.454	0.703
Mean	0.180	0.208	0.571	0.587
Isodata	0.068	0.142	0.117	0.748

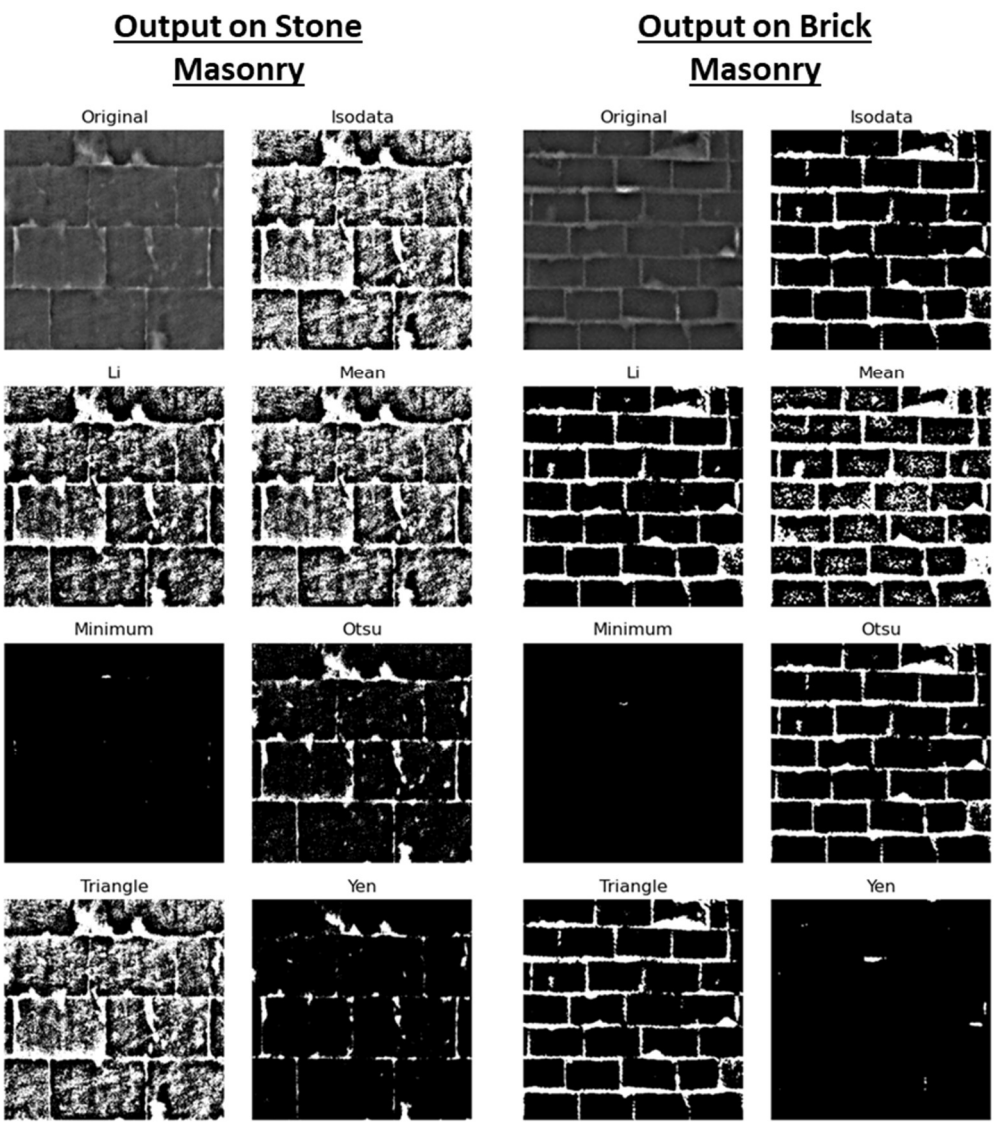


Figure 6.10: Threshold methods applied to a section of brick and stone masonry in tunnel T1.

6.5.2. Overall results

A method was selected from each of the 5 different segmentation strategies for evaluation on the test data. The methods are compared in this section using the IoU metric. Each method has different data preprocessing requirements that are outlined in Table 6.8 alongside a summary of the chosen method. The thresholding and Canny methods were applied directly to the testing data, while the machine learning methods were trained first on

the 10m long section of tunnel training data before being applied to the test data section. In order to avoid edge effects reducing performance at the sides of each image patch (Pielawski and Wählby, 2020), the CNN based methods were each evaluated three times on the testing data. On each application the image patches were offset by one third of an image patch size in the vertical and horizontal dimensions. A pixel was only classified as containing a joint if at least two out of three of the outputs produced a positive joint classification at the pixel.

The selected Otsu and Canny methods are compared against the machine learning based methods in Table 6.9. Figure 6.11 shows the output of each method on the test data. The Machine Learning methods all achieve substantially higher IoU scores than the traditional computer vision methods, although the traditional methods do have the advantage of not requiring training and having a negligible processing time when applied to the test data.

XGBoost identifies some joint locations more accurately than the Canny and Otsu methods, but a substantially lower IoU was achieved than for the other machine learning based methods. It can be seen in Fig. 4b that the algorithm could not distinguish between joints and other anomalous areas that effect surface roughness properties such as efflorescence and spalling. An additional downside of the XGBoost method was the substantial computation time on the test dataset, which was 150% to 200% more than the KPConv and U-Net methods. This was because the eigen features and surface roughness values had to be evaluated before the trained XGBoost model could be applied.

Table 6.8: Overview of selected segmentation methods.

Method	Cloud preprocessing requirement	Description	Relative cost of compute	Background
OTSU (Thresholding)	Tunnel point cloud must be unrolled and then rasterized into a 2D Image. Image contrast should be locally normalised.	2D image intensity thresholding: Uses a single threshold to segment classes	Low	Proposed by Otsu (Otsu, 1979), the method sets a threshold on the image intensity by minimising the intra-class variance.
CANNY (Edge detection)	Tunnel point cloud must be unrolled and then rasterized into a 2D Image. Image contrast should be locally normalised. Output needs post processing with erosions to correctly match the joint width.	2D image based edge detection workflow using gaussian smoothing and Sobel filters	Low	Developed for edge detection by (Canny, 1986).
XGBOOST (Decision Tree)	K Nearest neighbour points must be calculated within specified radius. 3D roughness features must be calculated for each point, as outlined by Hackel et al. (2016).	point feature based supervised machine learning: Boosting decision tree ensemble method	Medium	Initially released in 2014, the XGBoost library (Chen and Guestrin, 2016) was developed to improve the performance of decision tree methods.
U-NET (2D CNN)	Tunnel point cloud must be unrolled and then rasterized into a 2D Image. The image must be split up into patches to be input to the CNN in turn.	2D image supervised deep learning: 2D encoder-decoder CNN	Medium with GPU	Developed by Ronneberger et al., (2015), the U-Net was the first semantic segmentation network to show excellent transfer learning performance.
KPCONV (Pointwise CNN)	K Nearest neighbour points must be calculated within specified radius. The cloud must be split up into patches to be input to the CNN in turn.	3D point cloud supervised deep learning: 3D encoder-decoder CNN with adjustments to work with sparse point clouds	High with GPU	Thomas et al., (2019) developed KPConv in order to investigate how image convolutions could be adapted for sparse 3D point clouds.

Rigid KPConv did not achieve an IoU substantially superior to XGBoost, although the IoU improved when applied to the unrolled point cloud. This was likely because the network did not need to characterize features in as many orientations. Applying flexible convolutions also improved the KPConv IoU, as the convolutions were able to adapt to the single surface nature of the data. Despite the substantial training time and accurate segmentation in some areas, the IoU of the best performing KPConv network did not surpass the U-Net, as it failed in areas where the joints were very narrow. It is likely that the superior transfer learning ability of the U-Net is a key factor in its performance. While it is possible that a deeper version of KPConv would be better able to characterize the multi scale features, this may lead to overfitting given the limited training data. With the small volume of training data and limited amount of training data augmentations, none of the KPConv models could accurately segment the smaller brick joints. The U-Net forms the only model able to effectively segment the brick masonry and is also the fastest of the machine learning methods to train. It is possible that with additional data and more compute time, KPConv could achieve superior performance.

Table 6.9: Masonry joint segmentation performance on tunnel T1 test data

Network	Point cloud flattened	IoU	Training device	Training time (mins)
Canny	Yes	0.185	N/A	N/A
Otsu	Yes	0.198	N/A	N/A
U-Net	**Yes**	**0.558**	**PC1 (GPU)**	**34**
XGBoost	No	0.185	PC1 (CPU)	100
KPConv Flexible	No	0.337	ARC4 (GPU)	2880
KPConv Flexible	Yes	0.376	ARC4 (GPU)	2880
KPConv Rigid	No	0.247	ARC4 (GPU)	2880
KPConv Rigid	Yes	0.300	ARC4 (GPU)	2880

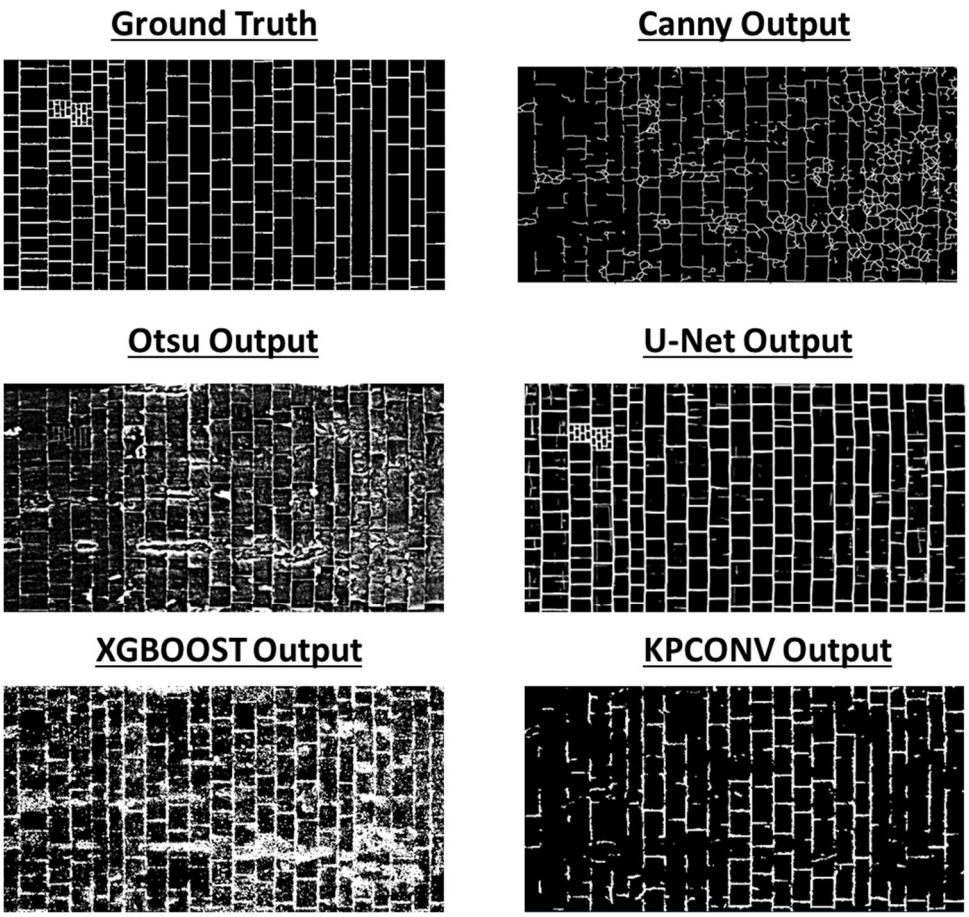


Figure 6.11. Algorithm outputs on a section of the test data compared with the ground truth.

6.5.3. Discussion

Since the U-Net was only model to achieve an IoU of greater than 0.5, further study focused on further improving the U-Net model and evaluating other architectures of 2D CNN model. Overall, the performance achieved by the U-Net model suggests that small scale surface features were equally well characterized when viewed as a 2D depth map rather than as the raw 3D data. It also confirms the validity of the unrolling and rasterization method selected for use in the spalling severity segmentation workflow proposed in Chapter 5. The U-Net performance shows that it is possible to train a CNN

to effectively segment masonry joints with limited training data and despite the distortions inherent in the generated 2D depth maps.

6.6. 2D CNN optimisation

In order to ensure that the optimum 2D neural network design was chosen, a variety of neural network architectures were assessed. Further to the original U-Net developed by Ronneberger et al. (2015), the UNet++ adaption by Zhou et al. (2018) and the transformer based SegFormer (Xie et al., 2021) were tested with different encoders as their backbone. While there are many architectures that achieve state of the art semantic segmentation performance in a range of domains, these networks represent state of the art U-Net and transformer based designs and are outlined in detail in the following subsections. The spalling severity segmentation workflow is designed such that other neural networks could replace these following future developments in semantic segmentation technology. This section aims to find the trained model that achieves the highest IoU on the T1 testing data to enable further analysis of the spalling severity segmentation workflow without joint segmentation causing a performance bottleneck. Each architecture was evaluated using a grid search with 3 factors, shown in Table 6.10. Other hyperparameters were fixed using values determined in section 6.4.1.

Table 6.10: CNN training parameters evaluated in section 1.6

Parameter	Values	Search type
Architecture	U-Net, UNet++, SegFormer	Grid
Encoder	VGG19, ResNet34, ResNet152, Mit_b5, Mobilenet_v2, ResNeXt50_32x4d, Xception, Efficientnet_b5, MobileOne_s4	Grid
Augmentations	All + Fixed size crop, All + Random size crop	Grid
Optimiser	AdamW	Fixed
Initial learning rate	0.0001	Fixed
Image dimension	384	Fixed
Pretrain weights	ImageNet	Fixed
Eps	0.0002	Fixed
Loss function	Soft Dice	Fixed
Batch size	8	Trial and error
Max training epochs	1200	Trial and error

6.6.1. UNetPlusPlus

The UNetPlusPlus 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 between the skip connections. A diagram is shown in Figure 6.12. The convolutions in the skip connections help to convert the feature map from the encoder to one better compatible with the format of feature maps found at the same level within the decoder. The convolutional layers within the skip connections are also densely connected, further adding flexibility to the network. Finally, by adding upsampling connections between the skip connection convolutions in different levels of the network, the network performs in effect as a series of nested U-Nets of varying depths. To take

advantage of this, deep supervision is used, where the output is taken from each step along the skip connection of the top layer of the network and averaged before the loss function is evaluated. In combination, these innovations should help the overall network better characterise features at different scales.

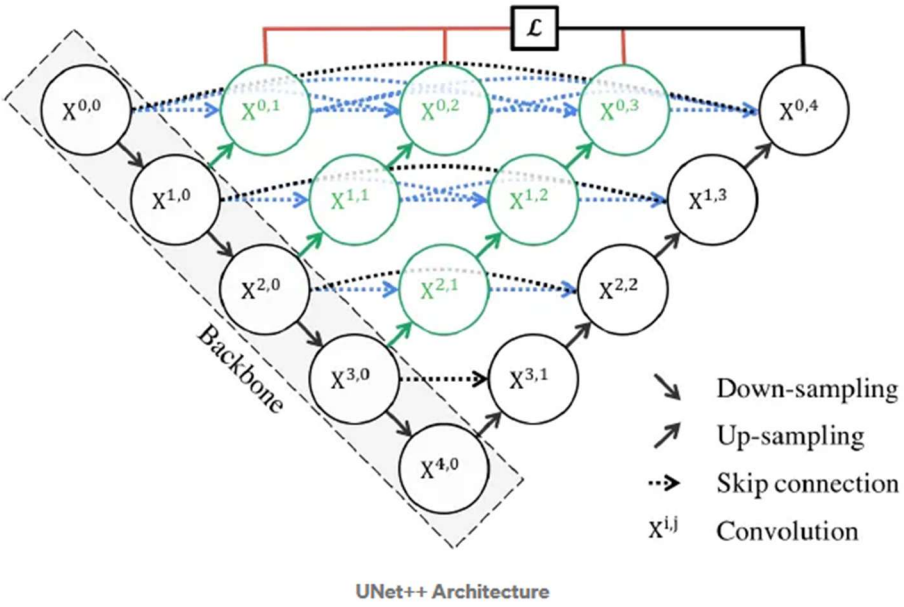


Figure 6.12: The UNet++ architecture from Kamrul Hasan and Linte (2019)

6.6.2. SegFormer

Developed for use with the transformer based Mit_Bx encoder series, the SegFormer model (Xie et al., 2021) builds upon the Vision Transformer (ViT) (Dosovitskiy et al., 2020) architecture to apply the transformer concept to semantic segmentation. Transformer models first developed by (Vaswani et al., 2017) introduce attention modules that weight the global importance of input features. This makes them better able to characterise long-range dependencies and contextual relationships between different regions of data. The SegFormer model splits the input image into 4x4 patches which are each given a positional embedding. Shown in Figure 6.13, the decoder of the

model uses standard multilayer perceptrons to fuse the output of the encoder and project them to segmentation logits. This makes use of the multi scale spatial information stored in the hierarchical encoder's outputs to directly generate the segmentation output, producing a more computationally efficient design than the standard skip connection and upsampling methods used in U-Net based decoder designs. While the SegFormer has been shown to achieve state of the art segmentation performance on benchmark datasets (Makarov et al., 2025; Sourget et al., 2024), more memory and compute power is required overall than similar depth U-Net style models due to the self-attention modules used within the Mit_Bx encoder.

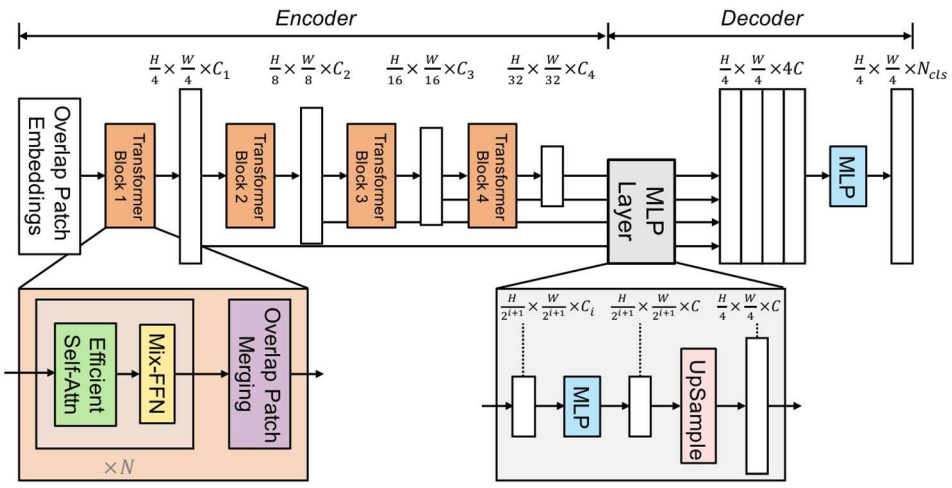


Figure 6.13: The SegFormer framework from Xie et al. (2021) showing the Mit_b4 hierarchical transformer encoder with an All-MLP decoder.

6.6.3. Encoders

Each architecture was trialled with each of the encoders described in Table 6.11 as the network backbone.

Table 6.11: Encoders evaluated in this section

encoder	No. of parameters	date	Description
VGG19 (Very Deep Convolutional Networks for Large-Scale Image Recognition)	20M	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 (Deep Residual Learning for Image Recognition)	21M	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.
ResNet152 (Deep Residual Learning for Image Recognition)	58M	2015	This is a deeper version of ResNet34 and is tested to evaluate the effect of using a network with more trainable parameters.
Xception (Deep Learning with Depthwise Separable Convolution)	22M	2016	Xception replaces the Inception modules within its predecessor, the Inception network, with depthwise separable convolutions.
ResNeXt50_32x4d (Aggregated Residual Transformations for Neural Networks)	22M	2017	Introducing the variable of cardinality, ResNeXt adds a split-transform-merge strategy to ResNet.
Efficientnet-b5 (Rethinking Model Scaling for Convolutional Neural Networks)	28M	2019	Efficientnet was developed as a scalable method that uses a compound coefficient to uniformly scale network depth and width with increased image resolution. It uses Inverted bottleneck residual modules and squeeze and excitation blocks.
Mit_b5 (Xie et al., 2021)	81M	2021	5 layer state of the art transformer-based encoder design. Adapts the vision transformer (ViT) encoder for semantic segmentation by adding multiresolution feature maps alongside other efficiencies.
MobileNet_v2 (MobileNetV2: Inverted Residuals and Linear Bottlenecks)	2M	2018	With only 2 million parameters, Mobilenet was developed to operate in low memory environments by introducing inverted bottleneck residual modules.
MobileOne_S4 (Vasu et al., 2022)	12M	2022	Designed to reduce latency at inference, MobileOne is a lightweight network that overparameterises during training with multiple branches. The network is then reparameterised to a single branch during inference.

All encoders were setup with parameters from ImageNet pretraining. A selection of encoders were evaluated with parameter numbers that were compatible with the amount of VRAM available on L2. Both ResNet 34 and ResNet 152 were evaluated in order to investigate the impact of the number of learnable parameters on performance. The lighter weight MobileNet_v2 and MobileOne_s4 encoders were evaluated to investigate the potential of conducting inference on lower performance devices. If these networks achieve adequate segmentation performance, then the workflow may be able to be applied onsite using typical portable computing devices, immediately after scanning. This paves the way for live onsite condition visualisations during tunnel inspections.

6.6.4. Data resizing and augmentation

The training images were augmented to increase the effective amount of training data. Applying augmentations can improve the generalisability of the method by widening the feature distribution of the training data. The parameters and probabilities of the transforms being applied were optimised using trial and error, and through visual comparison against abnormalities present in the available real datasets. This creates 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 (Buslaev et al., 2018):

- 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.

Before these augmentations were applied, the input images were cropped and resized during each training epoch. The input image patch dimensions were set as 512x512, which were then cropped at random locations down to the 384x384 dimension used as input to the neural network. Randomly cropping ensures that input features appear in a different location each time. This reduces overfitting by encouraging the network to learn the nature of each feature rather than just the location. This is important given the limited

volume of training data available. In addition to fixed size cropping, as part of the hyperparameter grid search, each network was also trained using random sized cropping, where crops were taken at dimensions varying between 300*300 and 450*450 before being resized to 384*384. These dimensions were obtained through trial and error. Resizing the image shrinks and expands lining features, further improving the CNN's generalisation capabilities to better characterise features with different sizes. An example of the augmentation workflow on a section of tunnel T2 is shown in Figure 6.14 and optimum augmentation parameters are shown in Table 6.12.

Table 6.12: Data augmentation parameters

Augmentation	probability	parameters
Random Sized crop	1	Min, Max dimensions = 300,450 Resized output dimension = 384
Horizontal Flip	0.5	-
Vertical Flip	0.5	-
Elastic transform	0.4	alpha=500, sigma=40, alpha_affine=8
Grid distortion	0.3	num_steps=5
Random Brightness	0.2	brightness_limit =0.5
Random Contrast	0.4	contrast_limit=0.2
Gaussian Noise	0.2	std_range= [0,0.05]
Perlin Noise	0.2	Variation = 0.3, Octaves = 2, persistence = 0.2, lacunarity = 2.0

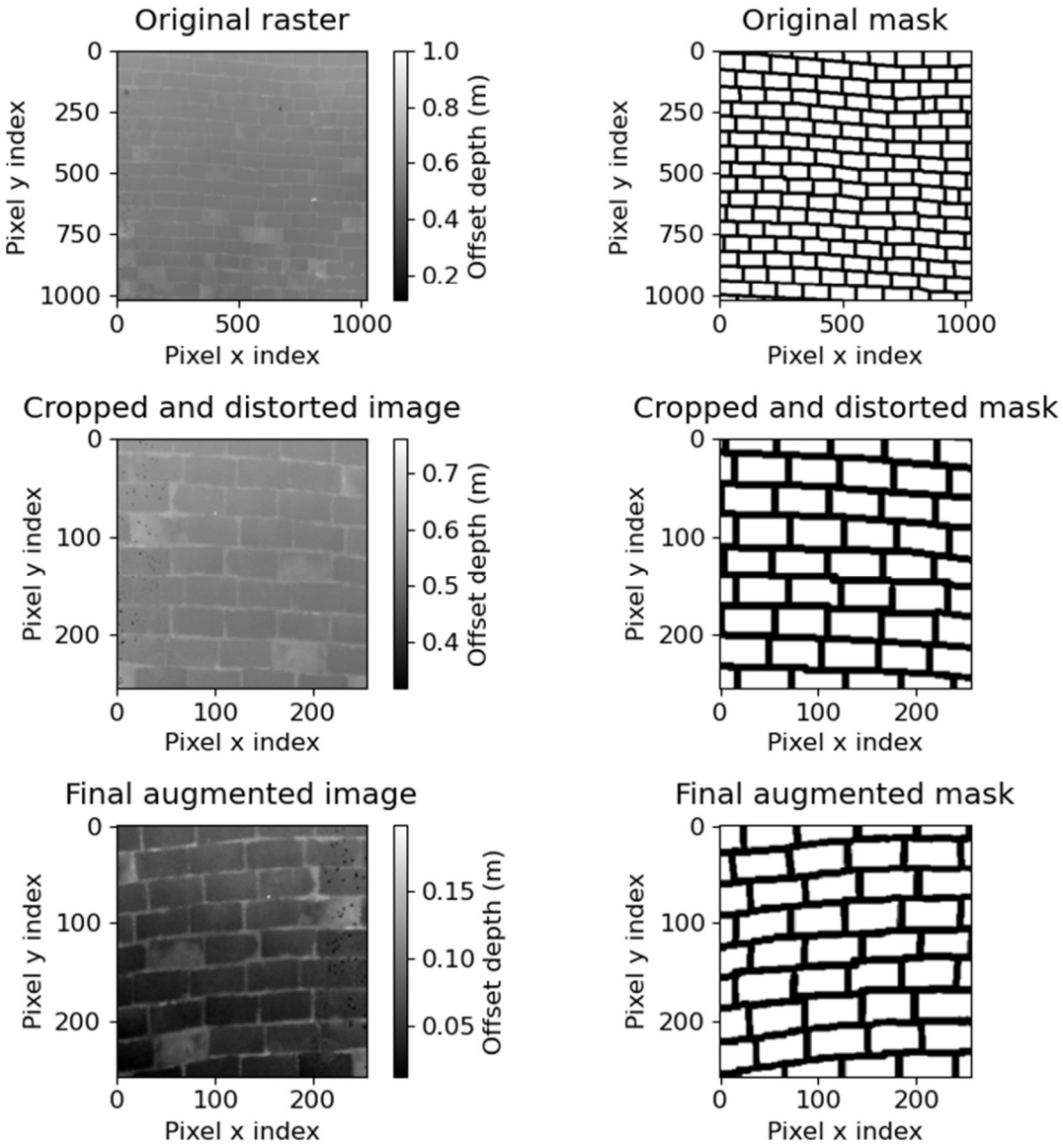


Figure 6.14: Crop and augment workflow with all possible augmentations applied to a patch of tunnel lining depth map.

6.7. Results

Before the optimum CNN setup is identified, the loss curves must be inspected to ensure that each network is converging and that the optimum number of training epochs has been chosen to prevent under or overfitting. It is expected that the training loss will decrease as far as the network is able to learn the features of the training data, so is dependent on the network's

knowledge capacity. The validation loss should show how well the network is learning transferable feature relationships and is indicative of unseen data performance. However, a low validation loss does not guarantee good test data performance, as the validation and training data samples are taken from different patches of tunnel within the same area, so are likely to have similar features. The testing data is taken from a different part of the tunnel, where lining features such as masonry geometries or damage types may be different. If underfitting is occurring, then the validation loss will still be decreasing at the last epoch. If the network is overfitting, then the validation loss will start to increase as the training loss continues to drop. Finally, if the network fails to converge then there will be no substantial drop in either the training or validation loss during the training. Using the U-Net with ResNet34 model with a fixed image crop size as the baseline, the loss curves when the architecture and backbone are changed are shown in Figure 6.15, alongside the loss curve when image resizing is used during the augmentation stage.

Changes in all three parameters of the grid search do not prevent the network from converging, although the network with the older VGG19 encoder shows a slower convergence. Each of the loss curves show relatively similar features, with the only important difference being more noise in the validation curve of the SegFormer model. This suggests that it is less stable and requires more data to effectively train, although the noise level is not significant enough to be a concern here.

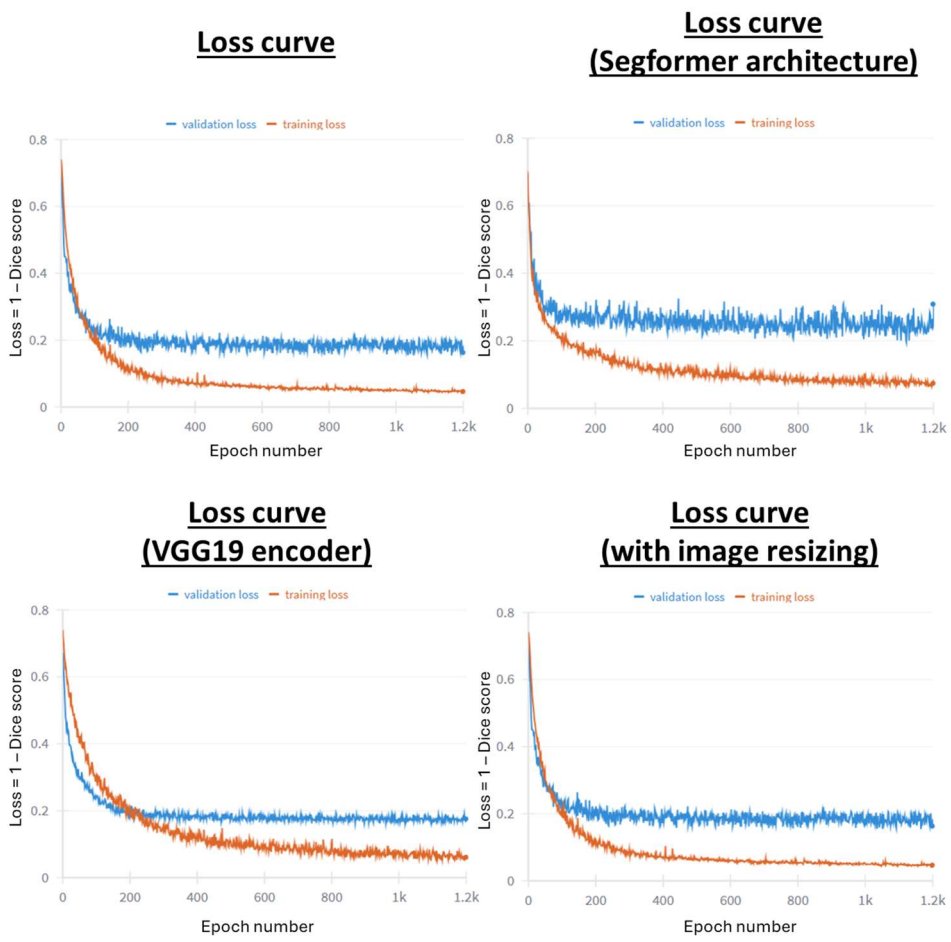


Figure 6.15: Training and validation loss curves for a U-Net model trained for 1200 epochs with a ResNet34 encoder using the ADAMW optimiser and fixed input image crop sizes. Loss curves are shown when the architecture and encoder are changed and when input images are randomly cropped and resized during training.

Figure 6.16 shows the maximum test data IoU performance achieved by encoder for a U-Net architecture model with no image resizing. The Mit_b5 transformer based encoder shows the highest performance. The number of training parameters show a slight positive correlation with the IoU. This is expected as a larger number of parameters results in a network with a higher knowledge capacity. Despite this, the lightweight Mobilenet_v2 and MobileOne_s4 networks still achieve adequate performances of 0.555 and

0.564 respectively. The date each encoder was designed also shows a small positive correlation with IoU as more recent encoders contain design innovations intended to make them more efficient and effective.

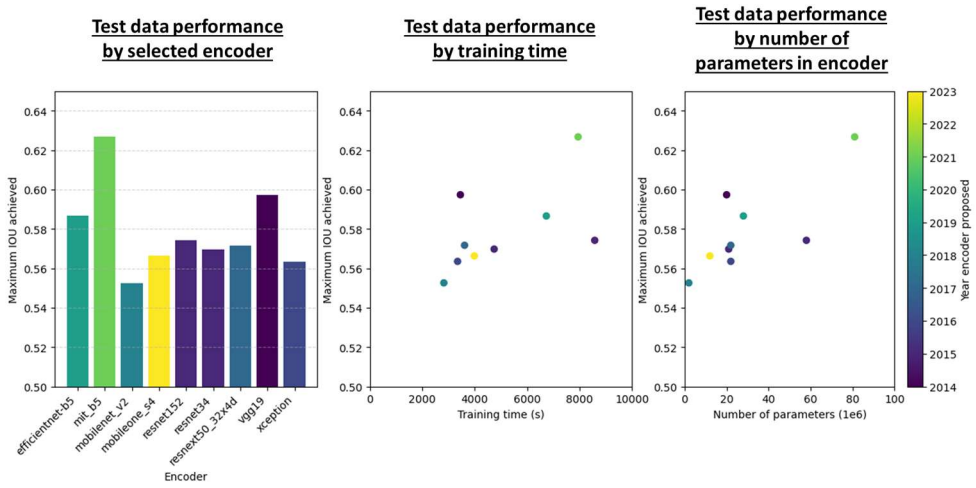


Figure 6.16: Test data performance for U-Net architecture showing the impact of chosen encoder and how the associated changes in training time and network size contribute to performance differences.

Figure 6.17 shows the performance by network architecture (showing performance with the most effective encoder for each network) and demonstrates the impact of image patch cropping and resizing on performance. For both the top performing encoder and network architecture, introducing image resizing improves performance. This is because these networks have a higher knowledge capacity for capturing features at different scales. Both the SegFormer model and the U-Net achieve their best performances with the Mit_b5 encoder. Their performances are similar showing that the use of self-attention modules in the encoder is the key driver of performance. The reason for the comparatively low maximum performance of the UNet++ network was because it did not converge when combined with the Mit_b5 encoder. The highest performing UNet++ network

used the ResNet34 backbone, showing that the architecture is less flexible than the U-Net to work effectively with more recently developed encoders.

The performance improvements achieved by the top performing Mit_b5 U-Net over the base U-Net ResNet34 design trained in section 1.4.1 is shown in Figure 6.18. The confusion matrix shows that both the proportion of false positive and false negative pixels are reduced, while the proportion of true positive and true negative pixels are increased for the new trained network. As a result, the Mit_b5 U-Net trained with image augmentations and image crop resizing was chosen as the masonry joint segmentation model to use within the spalling severity segmentation workflow.

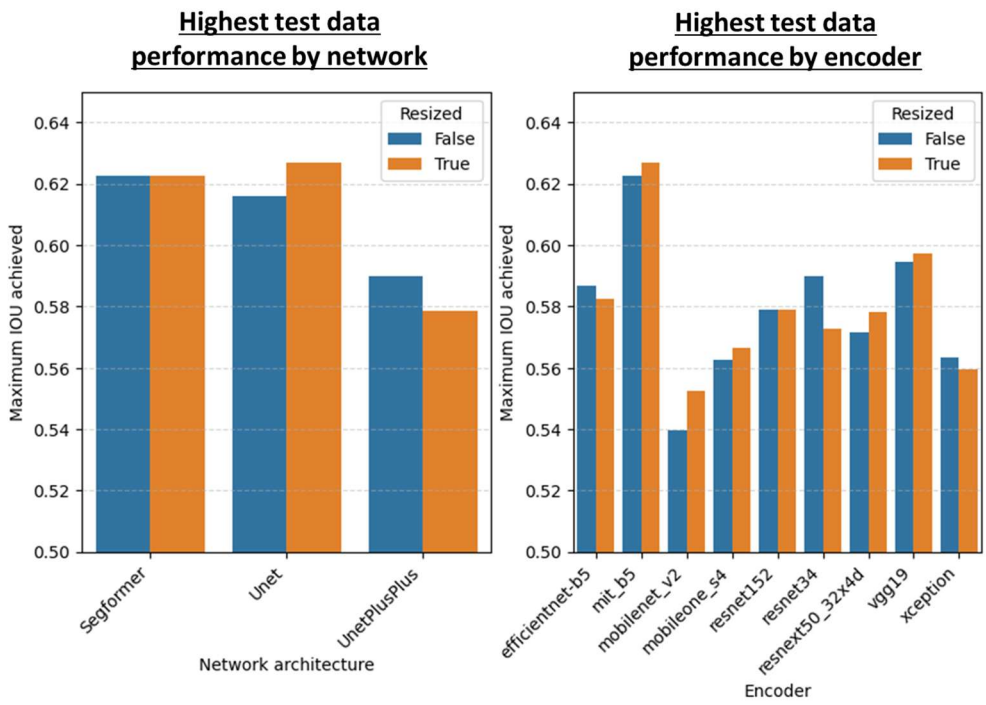


Figure 6.17: Maximum test data performance achieved for each architecture and encoder, showing the impact of input image patch resizing during training.

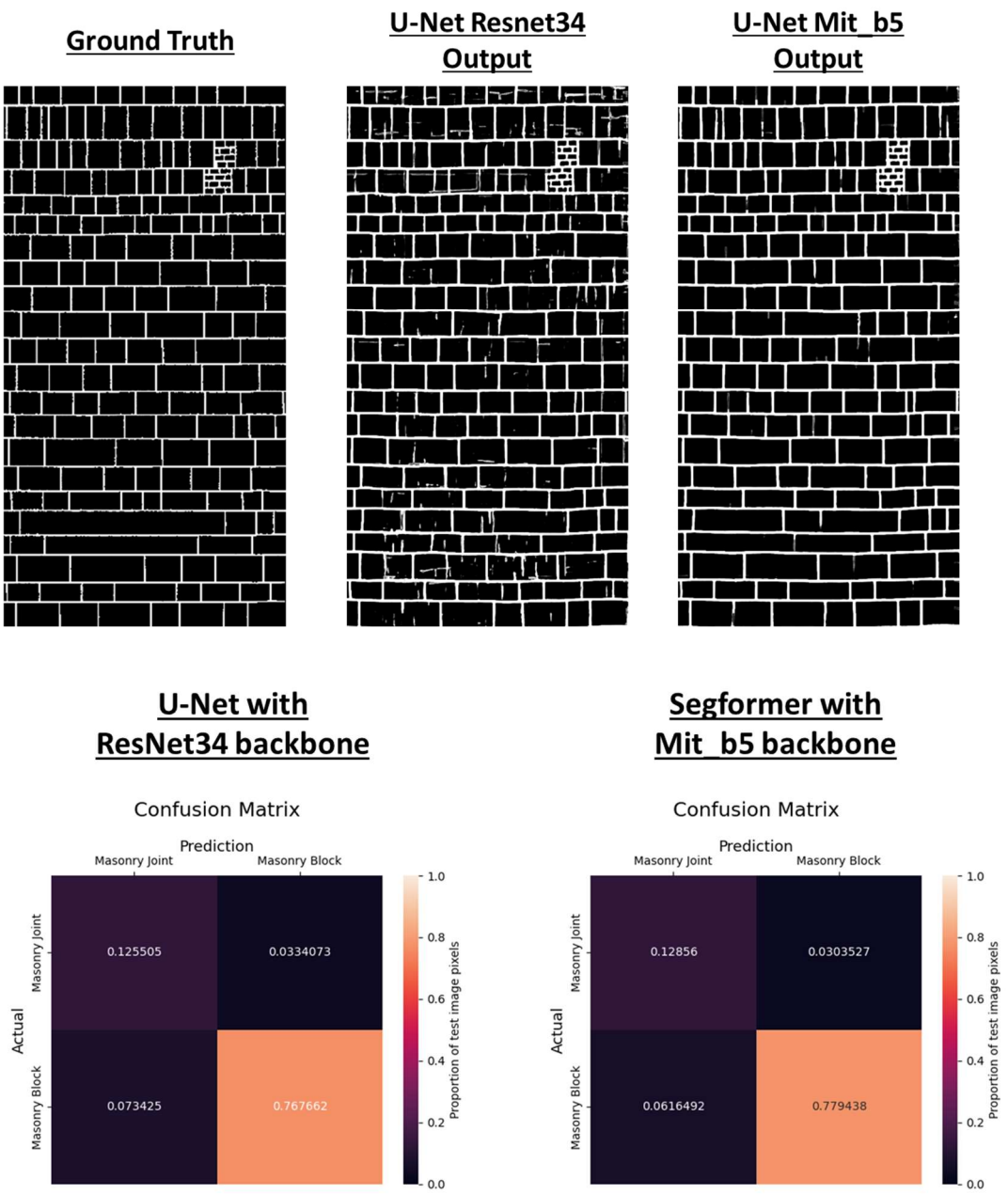


Figure 6.18: Confusion matrix and visual comparison for basic U-Net with ResNet34 backbone and selected U-Net model with Mit_b5 backbone

6.8. Damage segmentation

Given that 2D CNN category of methods achieved the highest IoU for joint segmentation and that the 2D depth map nature of the input data is the same for damage segmentation and joint segmentation, encoder-decoder style CNNs were selected for the masonry damage segmentation task. A similar grid search to determine the best performing hyperparameter and network architecture was conducted and a UNet++ with a MobileNet_v2 encoder achieved the best IoU of 0.459. 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 figure 6.19. The UNet++ model is shown to identify the general locations of the largest areas of spalling well, although the exact boundary of the damage is often incorrect. Damage segmentation performance is lower than joint segmentation performance but is qualitatively considered adequate due to the robustness of the overall spalling segmentation workflow to damage segmentation performance variation.

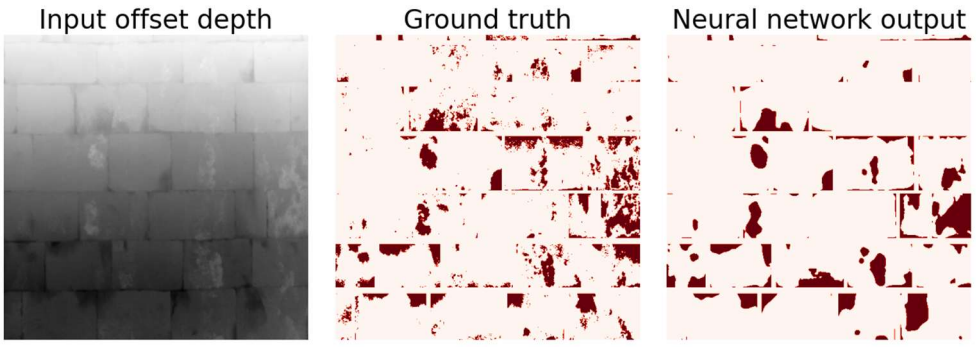


Figure 6.19: Section of masonry damage segmentation output on T1

6.9. Discussion and next steps

The aim of this chapter was to select a method to use within the joint and damage segmentation steps of the spalling severity segmentation workflow proposed in Chapter 5. This chapter evaluated multiple methods for masonry joint semantic segmentation from 3D point clouds obtained by lidar. A U-Net CNN with an Mit_b5 transformer-based encoder achieved the best performance by IoU, outperforming traditional computer vision methods, Decision tree and 3D neural network based approaches.

Of the traditional computer vision approaches, the Otsu method provided the best results and could be applied directly without training. However, even when input depth map rasters were preprocessed with gaussian differencing, the method only worked well in less noisy situations where the block faces clearly protruded from the mortar joints. Due to the use of a single threshold on depth, the Otsu method also misdefined deep spalling as joint locations. The XGBoost, decision tree-based method was also not able to distinguish joints from other lining surface abnormalities.

Of the CNN-based approaches the U-Net performed better than KPConv, demonstrating that reducing the dimensionality of the task to 2D improves performance given the limited availability of training data. A more comprehensive analysis of available 2D CNN models was then conducted. The Mit_b5 encoder was the most modern encoder assessed and given the extent of research into 2D supervised semantic segmentation methods in recent years, it is likely that it’s performance will be surpassed in the future. The proposed workflow was therefore designed to be flexible to replace the joint segmentation model with other semantic segmentation models as state of the art designs evolve. Nevertheless, there are further questions pertaining

to such segmentation model's real world application that will continue to be relevant regardless of the precise model applied.

While the U-Net and KPConv methods performed well on the stone tunnel lining case study, it is possible that performance will decrease when the method is applied to brick masonry. Further research was therefore conducted in Chapter 7 to determine how well CNN models can generalise to operate on tunnels with different masonry block materials and geometries and what are the most effective methods of training a model for real world tunnel condition assessments. Furthermore, for a CNN based method to be trusted for a tunnel analysis task, engineers would need a method of determining the uncertainty in a CNNs outputs on a specific tunnel without extensive manual verification. Chapter 8 aims to address this challenge.

7. Generalization analysis*

This chapter includes parts of published works: (J. Smith and Paraskevopoulou, 2025)

Although supervised training of Convolutional Neural Networks (CNNs) has generated excellent semantic segmentation performance in many applications (Dong and Catbas, 2021), their effectiveness for real world downstream tasks is not often assessed. Chapter 5 showed that accurate masonry block instance segmentation is required for the proposed automated masonry spalling severity assessment workflow to be effective. However, while the performance of the trained masonry joint segmentation models evaluated in Chapter 6 are promising, the connectivity of each masonry joint was often not accurately captured leading to poor block instance segmentation performance. There are often small gaps in the produced joint mask caused by poor image quality, narrow joints, obscured joints or masonry damage. Generating an accurate masonry model requires complete closure of joints to ensure that individual block instances can be isolated by a connected components analysis (Loverdos and Sarhosis, 2023; Smith et al., 2024). This necessitates the use of inflexible post processing methods to estimate where joints should be connected (Ibrahim et al., 2020; Smith et al., 2024). By investigating the topological accuracy of the trained CNN and training with a loss function that targets the joint connectivity directly, the block instance segmentation performance can be improved, and the method can be made faster and more generalisable.

There is limited knowledge on how a model should be trained for masonry joint segmentation given the typical real world tunnel analysis task where only limited training data may be available or affordable. For CNNs to

become a credible method of generating masonry structural models in industry, their reliability must be quantified. The wide variety of possible structural geometries and masonry types combined with the extent of damage typically present makes it challenging to develop a universally applicable automated masonry block isolation and labelling method. However, it is unknown how much performance variation can be expected even when a model is trained under similar conditions.

This chapter evaluates the ability of CNN-based masonry joint segmentation models to generalise to different tunnels. It conducts a statistical analysis to evaluate how masonry block instance segmentation performance varies with different training regimes and whether it can be improved by training with topological loss functions. and contributes to the field as follows:

- A practical comparison of loss functions for masonry block semantic segmentation is conducted.
- The impact of different training data combinations is studied to evaluate how neural networks should be applied on real tunnel analysis tasks.
- A new performance metric for masonry block semantic segmentation is proposed.
- The variability of segmentation performance is assessed.

7.1. Limitations of existing pixelwise methods

Small decreases in the pixelwise accuracy of masonry joint segmentation masks can lead to substantial errors in the produced spalling severity model if the identified joints do not have the correct connectivity. (Ibrahim et al., 2020) addressed this by post processing the deep learning output with a watershed algorithm to ensure joint connectivity (an approach later followed

by (Smith et al., 2023)) and using a Generative Adversarial Network (GAN) to infill obscured parts of masonry images. (Smith et al., 2024) took an alternative approach, trading flexibility for accuracy on masonry with regular joint orientations, by statistically fitting vertical and horizontal lines to the joint segmentation output. While these approaches successfully improve the joint connectivity, they require manual tuning of algorithm parameters on each application and there has been limited research assessing the final block instance segmentation performance.

The issue of ensuring segmentation connectivity is not unique to masonry block detection. The analysis of networks is a common goal of many computer vision studies, from modelling traffic flow in road networks to measuring land usage (Yuan et al., 2021). The end goal is the identification of network properties, such as the connectivity of different locations or the number of enclosed regions. These properties constitute the Image topology; The fast-moving field of topological image analysis focuses on understanding spatial relationships of structures in Images (Singh et al., 2023; Wasserman, 2018). While pixelwise segmentation accuracy is important for assessing the validity of these studies, the exact location of boundaries is often of less importance compared to the identification of topological features.

Popular semantic segmentation architectures such as the U-Net are typically trained to minimise total pixel or segmentation error in these tasks and do not consider image topology. Minimising variants of cross-entropy or soft dice loss functions is standard (Xu et al., 2023) and was previously applied for masonry joint segmentation. Recent developments integrating topological analysis techniques into deep learning workflows offer substantial potential to improve the topological accuracy of these segmentations (El Jurdi et al., 2021; Zia et al., 2024).

Consequently, in order to improve joint connectivity and achieve more reliable identification of individual masonry block instances, this chapter investigates deep learning-based masonry joint segmentation methods from a topological perspective. Topological loss functions were applied during neural network training to encourage correct joint connectivity. A new metric that holistically assesses masonry block instance segmentation was developed to better understand how topological and pixelwise errors impact performance. This metric, the Blockwise Intersection over Union score, was then used to evaluate the suitability of these deep learning methods for automating end to end damage identification workflows in real world scenarios. Since there is limited research assessing how deep learning could be applied in practical condition assessments, recommendations are provided on how masonry block segmentation should be conducted given different volumes of available training data.

7.2. Methodology

Input 3D tunnel lining point clouds were first unrolled and rasterised into 2D images using the workflow outlined in Chapter 5. Analysis was conducted to investigate how masonry block segmentation performance is affected by training a CNN with ClDice instead of pixelwise loss functions in different implementation scenarios. Statistical analysis was then conducted to determine the repeatability of the results and inform a reliable procedure for masonry block segmentation. A flowchart visualising the full procedure is shown in Figure 7.1.

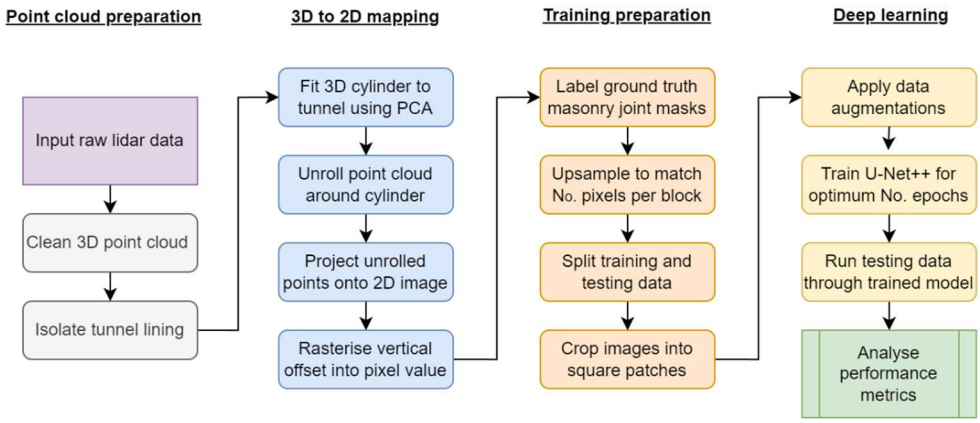


Figure 7.1: Outline of research procedure

7.2.1. Datasets

Lidar data from the stone lined tunnels T1, T3 and the brick lined tunnels T2 and T4 were used for the analysis within this chapter. These masonry-lined railway tunnels are located in the UK and managed by Network Rail, the UK's railway infrastructure manager. All were in reasonable states of repair, containing no damages requiring immediate remediation and are located on operational railway lines. Spalling of masonry and minor cracking was present in all tunnels in varying quantities. A brief summary of each tunnel is given in Table 7.1; background information on each tunnel is provided in Chapter 4.

Table 7.1: Datasets analysed from 4 different tunnels

Tunnel	Lining type	Cross-section dimensions width x height (mm)	Length (m)	Arch shape	Notes
T1	Stone	8680x5880	67	Horseshoe	Masonry in moderate condition. Some areas infilled with brick.
T2	Brick	8470x5660	145	Circular arch on curved sidewalls	Originally stone lined, largely relined with brick. Brick in good condition.
T3	Stone	9078x8094	138	Elliptical	Masonry in good condition.
T4	Brick	8427X5395	215	Circular	1.5m wide section of crown repaired with sprayed concrete. Brick in good condition.

KEY: = Tunnel 1 (Stone) = Tunnel 2 (Brick) = Tunnel 3 (Stone) = Tunnel 4 (Brick)

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.

Pretraining and augmentations were applied within the work outlined in this chapter. Synthetic data was not used so that the impact of real world data volume on performance could be clearly assessed without introducing volume and quality of synthetic data as additional factors in the analysis.

Representative sections of each point cloud were chosen for analysis. Sections for training containing at least 345 bricks were chosen away from sections for testing which had a least 55 bricks. 345 Bricks was chosen as a realistic number that an assessor could accurately label within 4 hours and for T1 forms the same 10m long section used for training the models in Chapter 6. This simulates a situation where the complete labelling, training and inference workflow could be conducted within 1 day for a single 20 m long tunnel. It was necessary to down sample the collected point clouds to enable a better comparison between tunnels and prevent the impact of point density reductions away from the lidar scanner location.

Multiple factors affect the original point density of the lidar scan. This includes different scanners being used and different scan speeds being applied based on the length of time available on site for the survey to take place. As a result, different datasets produce images at different levels of zoom depending on the point density, leading to differing numbers of pixels per brick. As shown in Table 7.2, the masonry in each tunnel also has different typical dimensions. The bricks in tunnels T2 and T4 have a relatively small variation in brick sizes and geometries compared to the stone lined tunnels T1 and T3. To help the trained CNNs generalise between the different datasets, the tunnels with fewer average pixels per masonry block were upsampled such that the average number of pixels per block was within a 10% range between datasets. The resulting image dimensions for the training and testing datasets are shown in Tables 7.3 and 7.4 respectively.

Table 7.2: Masonry properties of selected datasets

Tunnel	Typical masonry dimensions (mm)	Pixels per block	Scaling factor applied	Scaled Pixels per block
T1	680x360	8362	1	8362
T2	230x75	5674	1.20	8171
T3	500x200	3445	1.55	8277
T4	240x85	8467	1	8467

KEY: = Tunnel 1 (Stone) = Tunnel 2 (Brick) = Tunnel 3 (Stone) = Tunnel 4 (Brick)

Table 7.3: Training data properties

Tunnel	Number of blocks	Image dimensions width x height (pixels)
T1	345	1100x2900
T2	357	1780x1660
T3	354	1240x3022
T4	348	3100x1000

KEY: = Tunnel 1 (Stone) = Tunnel 2 (Brick) = Tunnel 3 (Stone) = Tunnel 4 (Brick)

Table 7.4: Testing data properties

Tunnel	Number of blocks	Image dimensions width x height (pixels)
T1	66	1040x590
T2	65	775x604
T3	64	1147x527
T4	59	1120x435

KEY: = Tunnel 1 (Stone) = Tunnel 2 (Brick) = Tunnel 3 (Stone) = Tunnel 4 (Brick)

7.2.2. Image preparation

Training data images were split into 512x512 patches. 20% of the patches were assigned as validation data, with the remainder used as training data. During training and validation, images were augmented to increase the quantity of data and improve the generalisation performance as follows:

- A random crop of 384x384 was made of each image.
- Random vertical and horizontal flips were applied, each with a 50% probability.
- Random brightness and contrast adjustments were applied with a 20% probability each, brightness limit of 0.5 and contrast limit of 0.5.

More augmentation types and higher probabilities of application increases the level of randomness in the training. This can lead to regularisation and improved generalisation performance, but also reduced training data performance and longer training times to convergence. The selected augmentations were chosen, as they were found to improve test data performance without substantially increasing the variability in performance between similar training runs. This enables us to clearly investigate the impact of loss function choice on test data performance.

7.3. Performance evaluation

Neural networks are evaluated by running a testing dataset through the trained network. The testing set should be representative of a real-world application of the neural network. The output performance must then be evaluated using a performance metric that captures the differences between the features of the ground truth and the neural network output that are important to the application domain. In the proposed application of automating masonry joint segmentation to document block geometries and

locations, it is key that the joint segmentation fully encloses each masonry block by respecting the joint topology in addition to having an acceptable pixelwise accuracy.

7.3.1. Existing metrics

While the IoU outlined in Chapter 6 shows segmentation quality with respect to exact pixel locations, it does not consider the overall shape of the segmented objects and the masonry joint connectivity. Persistent homology based methods analyse the topology of an image segmentation by studying the connectivity of segmented structures over a range of thresholds (Edelsbrunner and Morozov, 2014). The persistence is the range of thresholds that can be applied before the connectivity of an object is changed and can be plotted on a persistence diagram. If the output probabilities of a CNN are set as pixel values in an Image, then the persistences of objects in the segmentation can be evaluated. For a given threshold, the number of connected components and holes is called the Betti number which is often used as a simplified description of an Image's topological properties. The difference in Betti number between ground truth and predicted segmentations, the Betti error, can be used as a holistic measure of the topological errors in the segmentation. This measure does not consider the matching of topological features, only the total number, so a low Betti error does not necessarily indicate that each masonry block is segmented with the correct topology.

7.3.2. BwIoU

The metrics identified in section 3.3.1 can be used to assess the pixelwise and topological performance of masonry joint segmentation, but they do not directly describe individual block instance segmentation performance. For

the task of block documentation, identifying and labelling individual block instances is conducted using a connected component analysis performed using the 8-way connectivity Spaghetti labelling algorithm (Bolelli et al., 2020). The instance segmentation performance of the output of a connected components analysis should therefore also be assessed. We propose a Blockwise Intersection over Union metric (BwIoU). The BwIoU evaluates performance based on both the matching of predicted and ground truth block instances and the pixelwise segmentation performance of each match. The calculation of BwIoU is visualised in Figure 7.2 with the following steps:

1. An 8-way connected components analysis is run on the detected blocks to isolate predicted block instances. This is implemented using the OpenCV library (Bradski, 2000).
2. The location of the centroid of each ground truth and predicted block is calculated. The distance to nearest neighbour predicted centroids are calculated for each ground truth centroid and the closest predicted block is matched to each ground truth block.
3. Unmatched predicted blocks are discarded and the IoU is calculated between each ground truth block and its matched prediction. The mean of these IoUs over all blocks is calculated to produce the BwIoU.

Figure 7.2 focuses on the location of 2 ground truth masonry blocks and shows a hypothetical output of a CNN trained for masonry joint semantic segmentation. Running the 8-way connectivity connected components algorithm produces three predicted block instances as shown in stage 1. It is then necessary to match each of these to a ground truth block. However, since there are only two ground truth blocks, one of them must be discarded. The matchings are chosen based on the distances between the centroids of

the predicted and ground truth block locations. Each ground truth block is assigned a single predicted block. In the case shown, predicted blocks 1 and 3 are assigned to ground truth blocks A and B, respectively. Predicted block 2 is unmatched and discarded. Finally, in stage 3 the IoU is calculated per ground truth block against it's matched predicted block. The IoUs are then averaged across all of the ground truth blocks to produce a final BwIoU value. As shown in the diagram, although the IoU of block A is reduced by the low precision of prediction 1, The IoU of block B is penalised further due to the incorrect topology of its segmentation. Multiple common block segmentation errors are shown in Figure 7.3. These were used to verify the BwIoU in different situations.

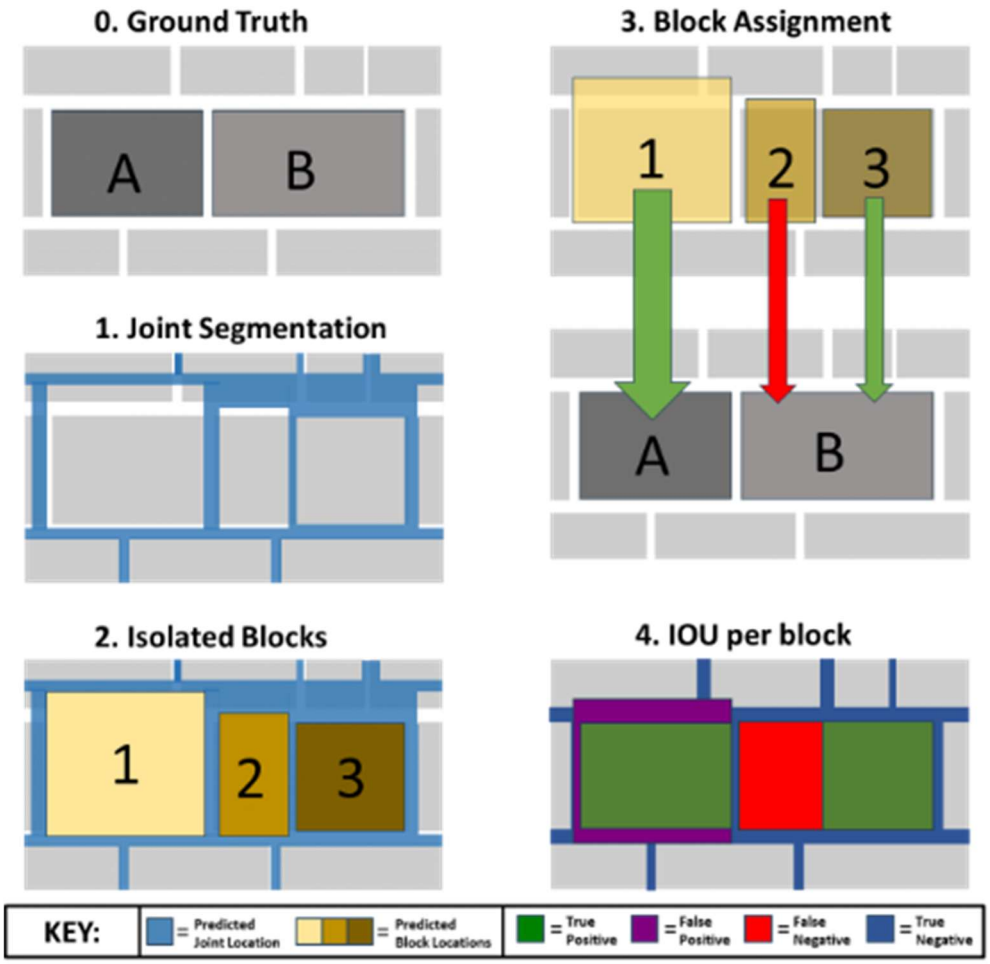


Figure 7.2: Visualisation of BwIoU calculation workflow

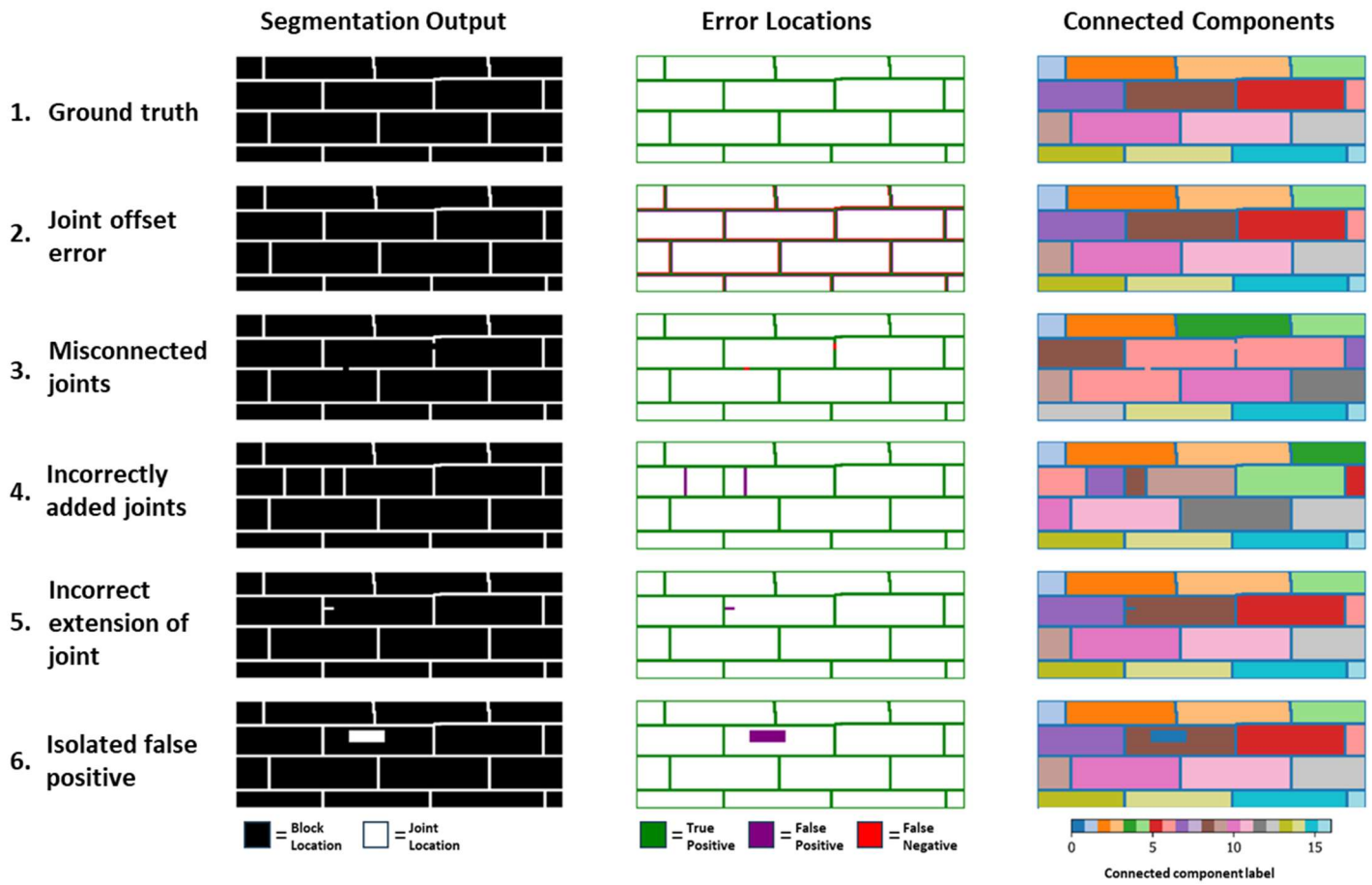


Figure 7.3: Visualisation of commonly observed segmentation errors

A comparison of metrics calculated for each of the hypothetical segmentations visualised in figure 7.3 are shown in Table 7.5 and discussed as follows:

1. **Ground Truth:**

   If a perfect segmentation is performed, then the BwIoU achieves its theoretical maximum value of 1.

2. **Joint offset error:**

   This common error occurs due to the difficulty in defining an exact boundary for each masonry joint, so the real or predicted joint location may be different to the Ground Truth which is drawn with constant width joints. In this case, the Joint IoU is substantially more affected than the BwIoU. Because the joint connectivity is correct, all blocks are correctly matched. Each block has a larger number of pixels than each joint, so a boundary error has a proportionally larger impact on the joint IoU than the block IoU.

3. **Incorrectly connected joints:**

   While only caused by a small number of pixel errors, the connectivity of 2 joints is broken. This causes the connected component analysis to detect only 1 block in a location where 3 should be present. As a result, the Betti number is 2 and the BwIoU is considerably reduced. The Joint segmentation IoU is only marginally reduced by the false negative pixels.

4. **Incorrectly added joint:**

   Here we see 2 extra blocks being incorrectly isolated due to the additional joints segmented resulting in a Betti error of 2. Similarly to situation 3, the joint IoU only has a small decrease, while the BwIoU is more substantially affected. However, in this situation the BwIoU is

decreased less than for situation 3. This is because for each additional joint, the area of masonry impacted is limited to a single block, while each missed joint causes two adjacent blocks to be merged during the connected component calculation causing a larger area of incorrect instance segmentation.

5. **Incorrect extension of joint:**

   This is where a CNN has predicted an additional Joint area which does not change the joint topology. This commonly occurs when a masonry crack is misidentified. Both the IoU and BwIoU are reduced due to the small number of false positives.

6. **Isolated false positive segment:**

   This error produces a single topologically incorrect island, as shown by the Betti error of 1. However, as a connected components analysis still produces correct block matchings, the BwIoU is only impacted by the small number of false negative block pixels. The Joint IoU is affected more due to the class imbalance.

Overall, it can be seen that the IoU of joint segmentation does not distinguish topologically incorrect regions. This is demonstrated with situation 3 producing a higher score than situation 2. The BwIoU, however, is strongly affected by incorrect joint connectivities and shows a decrease from situation 2 to 3. In the case of situations 5 and 2, where small and large segmentation errors are present respectively, the BwIoU score is penalised proportionally. However, as the block topology is correct, the BwIoU receives only a small decrease compared to the IoU for joint locations.

This shows that the BwIoU performs as expected in common situations, so is an appropriate metric for measuring masonry block instance segmentation performance.

Table 7.5: Comparison of performance metrics in different situations highlighting where IoU and BwIoU differs

Error type	Topological error	Precision	Recall	Accuracy	Proportion of matched blocks	Betti error	IoU of joints	BwIoU
1. None	No	1.000	1.000	1.000	1.000	0.000	1.000	1.000
2. Joint offset	No	0.696	0.696	0.917	1.000	0.000	**0.534**	**0.902**
3. Misconnected joints	Yes	1.000	0.993	0.999	0.867	2.000	**0.993**	**0.782**
4. Incorrectly added joints	Yes	0.967	1.000	0.995	1.000	2.000	**0.967**	**0.904**
5. Incorrectly extended joint	No	0.994	1.000	0.999	1.000	0.000	0.994	0.990
6. Isolated false positive	Yes	0.919	1.000	0.988	1.000	1.000	0.919	0.966

KEY: = High Quality = Medium Quality = Low Quality **Bold** = Difference in relative performance between IoU and BwIoU

7.3.3. Topological loss functions

Having specified a problem specific quality metric, this study investigates the impact of loss function selection on neural network performance. Specifically, it examines whether loss functions designed to optimise semantic segmentation topological correctness leads to better block instance segmentation performance. Beyond the Dice loss introduced in section 6.4.1, multiple other pixelwise loss functions have been proposed.

Binary Cross Entropy (BCE) is a commonly applied loss function across deep learning applications. In the case of binary segmentation, for a positive ground truth pixel it is equal to the logarithm of the probability of a positive segmentation. For a negative ground truth, it is equal to the logarithm of the probability of a negative segmentation. The total loss is the sum of the values for each pixel. BCE maximises the number of true positives and true negatives equally. However, it fails to take account of class imbalances, biasing predictions towards the larger block class in the case of masonry joint segmentation.

The Tversky loss developed by (Salehi et al., 2017) adapts the Jaccard loss outlined in Chapter 6 by adding weightings to the False Negative and False Positive terms used for calculation of the intersection. This enables manual tuning of the loss function to balance over and under segmentation. The Focal Tversky loss (Abraham and Khan, 2019) returns the Tversky loss to the power of another tunable parameter, γ. γ can be adjusted to either give a larger weighting to lower performance regions, improving convergence or it can be decreased to encourage the network to focus on challenging locations such as thin boundaries that have a relatively small impact on the overall IoU. Well tuned, Focal Tversky loss was shown to improve segmentation performance of narrow structures.

However, these loss functions do not consider the topological accuracy of the output segmentation and so do not directly seek to improve the BwIoU. Since the connected components step of BwIoU calculation is not differentiable, it is not possible to trivially adapted it into a task specific loss function. Consequently, other methods need to be applied to encourage topological accuracy.

One of the first studies to integrate topological information into a loss function for training deep learning semantic segmentation models was (BenTaieb and Hamarneh, 2016). They studied methods of improving an FCN for histology gland segmentation. They integrated smoothness and topological priors that were handcrafted around the known physical layout of glandular tissues. This generated up to 10% improvement of the Dice score compared to a per pixel loss formulation.

(Mosinska et al., 2017) created a more general loss function for segmentation of curvilinear structures by comparing the feature maps at different layers of a pretrained VGG19 network with the output segmentation. This harnesses the knowledge that the pretrained network will contain information about common object structures. With long thin structures common in the feature maps, the loss function will encourage these structures in the output segmentation, improving connectivity.

More flexible approaches using persistent homology methods have since been developed that penalise incorrect connectivity of image segmentations. Using a variant of the Wasserstein distance, a measure of the difference between two persistence diagrams, (Hu et al., 2019) created Topoloss. This was shown to consistently achieve a lower Betti number error compared to dice loss when applied with a U-Net to a variety of standard datasets. However, the complexity of this method is cubic to image size, so computation times can be unreasonably long when larger images are used.

Since then, (Shit et al., 2020) created the CenterlineDice loss (clDice) which has become one of the most popular methods for encouraging topological accuracy. They calculate the precision of the intersection of the skeleton of the ground truth with the segmentation prediction, and the precision of the intersection of the skeleton of the prediction with the ground truth. The clDice

is then the Dice score between these metrics. In order to calculate the image skeletons, they propose a soft skeletonization approximation using iterative min and max poolings to erode and dilate the images. This method shows excellent performance in a range of studies and forms the least computationally challenging connectivity focused loss function described here. As a result, it was chosen for this study.

(Hu, 2021) also went on to develop the warping loss. Image warpings are transformations that do not adjust the image topology. By conducting image warpings between the segmented and ground truth images, any topological errors will be manifested in critical points. These points cannot be adjusted without changing the image topology, preventing a full warping between segmentation and ground truth. The number of critical points is used as a measure of the level of topological inaccuracy. While this is a more flexible measure with a lower computational cost than the Topoloss, the approximations used do not guarantee topological consistency. In addition, the calculations involve substantial single thread CPU usage, leading to long runtimes despite the lower complexity.

As these topologically informed loss functions can be significantly impacted by small pixelwise differences, convergence of the neural network can be unstable during training. In order to ensure a smooth convergence, topological loss functions should be combined with pixelwise losses. Before practical application, the topological loss function proportion, α, should be assessed to determine the optimal loss function combination. A comparison of the discussed loss functions is shown in Table 7.6.

Table 7.6: Loss function comparison.

Function	Target feature	Computational cost	Origin
BCE	Pointwise accuracy	Low	-
Soft Dice	Dice score	Low	(Milletari et al., 2016)
Focal Tversky	Dice score	Low	(Abraham and Khan, 2019) modified the original Tversky loss by (Salehi et al., 2017)
Topo	Topological accuracy - Wasserstein distance	Very High	(Hu et al., 2019)
Warping	Topological accuracy – Critical points	High	(Hu, 2021)
ClDice	Topological accuracy – intersection of skeletons	Medium	(Shit et al., 2020)

7.3.4. Training configuration

The analysis code was developed in Python and deep learning was conducted using the PyTorch library (Paszke et al., 2017). In order to focus on the impact of loss function choice on segmentation performance, the same network design and hyperparameters were used for each training run. These are outlined in Table 7.7 and were chosen based on a grid hyperparameter search for a model trained and tested over all 4 tunnels with a soft Dice loss function. This included trialing a variety of network architectures and backbones taken from the Segmentation models PyTorch library (Pavel Iakubovskii, 2019). Each network was trained using the PyTorch framework for a maximum of 1000 epochs on the ARC4 machine outlined in Chapter 4. Each model was frozen at the epoch with the minimum validation loss (assessed every 20 epochs) and the performance metrics outlined in section 3.2 were evaluated on the frozen model.

Table 7.7: Chosen neural network architecture and hyperparameters

Architecture	Backbone	Optimiser	Weight decay	Initial learning rate	Image patch dimension
UNet++	ResNeXt50_32x4d	AdamW	0.002	0.001	384x384

7.3.5. Training scenarios

The first step to assess ClDice performance involves finding the optimum proportion of ClDice to combine with soft Dice within the domain. Following the guidance from (Shit et al., 2020), values of α from 0.1 to 0.5 were trialled. The UNet++ was trained for masonry joint segmentation using each α value. In order to evaluate the likely performance of each of these values in a real world setting, where condition assessment engineers may not be able to extensively tune neural networks for every dataset, it is vital that the distribution of likely performance values is analysed. Each training run was therefore repeated 7 times using different random seeds in order to verify the results and determine the variability of each training scenario. All four datasets were used for training.

Similar to this procedure for optimising ClDice proportion, weightings of the Focal Tversky parameters were tuned to optimise the average IoU over all datasets with the model also trained across all datasets. The parameters were then fixed for all further analysis.

After α was determined, analysis was conducted to assess the performance of each loss function in expected real world applications. BCE, Focal Tversky, Dice and ClDice loss functions were each used in combination with 5 different training regimes. Again, 7 repeats were used, producing 140 training runs for each of the 4 target testing tunnels and a total of 560 training

runs overall. This enables further statistical analysis of each method to assess whether observed performance differences are likely to be significant (Rainio et al., 2024).

Each training regime represents a possible real world scenario where masonry block identification may be part of a programme of work to create tunnel information models or conduct digital condition assessments. For each situation, different volumes of training data may be available. Each training regime is summarised in Table 7.8 and visualised in Figure 7.4. Each tunnel is split into separate training and testing regions so that the training data from each tunnel comprises a different section of lining than that used for testing on the same tunnel. By ensuring that all of the test sections are independent of the training sections, there is no data leakage.

In the first scenario, where data from only a single tunnel is available (Single), the neural network is trained on a small section of tunnel, then applied to a different section of the same tunnel. In the case of longer tunnels, having an assessor label masonry block locations over a small section which is then used to conduct an automated analysis of the rest of the tunnel may still be faster than an assessor manually labelling the whole tunnel.

The second application type is when a model trained on one tunnel is applied ‘off the shelf’ to an unseen structure (Other_all). Further variants of the second method are also assessed, including where training is only conducted on tunnels with the same lining type (Other_st). These are ideal scenarios, as no additional tunnel specific labelling is required.

The final methods consider the situations where data from other tunnels are combined with target tunnel data (Full_all), training the model on all available data. Variants utilising only data from tunnels of the same lining type are also assessed (Full_st).

Table 7.8: Comparison of training regimes

Label	Description
Single	The model is trained on one tunnel and tested on the same tunnel.
Other_all	The model is trained on three tunnels and tested on a fourth unseen tunnel.
Full_all	The model is trained on all four tunnels and tested on one of them.
Other_st	The model is trained on a tunnel with one lining material and tested on an unseen tunnel with the same lining material.
Full_st	The model is trained on both tunnels with the same lining material and tested on one of them.

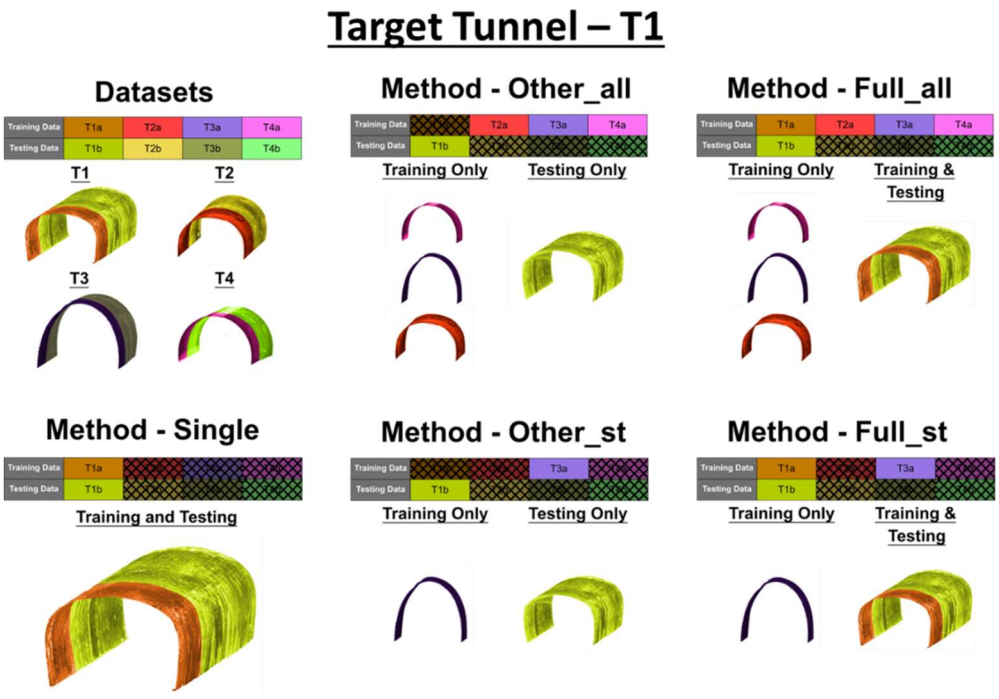


Figure 7.4: Visualisation of the 5 training regimes when applied to assess Tunnel T1.

7.3.6. Statistical analysis

The validity of conclusions about the impact of different loss functions and training regimes must be analysed in the context of natural variability between trained CNNs caused by the stochastic nature of neural network optimisation. Tukey's honestly significant difference (HSD) test (Tukey, 1949) was used to test whether there is likely to be a significant difference between the means of two compared analysis groups. A positive result indicates that the data samples are likely drawn from different distributions. This means that performance differences are not occurring solely due to randomness during training and changes in the training conditions or loss function are likely impacting performance. Tukey's HSD is a variant of the student's t-test (STUDENT, 1908) that corrects for the family-wise error rate and so can be used reliably when comparing multiple pairs of data groups. A significance level of 5% was chosen for the performance of two groups to be considered likely to be significantly different.

7.4. Results and Discussion

7.4.1. Optimal loss proportion

Since stone tunnel linings typically have a wider variety of block geometries than brick linings, performance of each loss proportion is evaluated by lining type in addition to a combined average score over all tunnels. Stone lining refers to the average performance over T1 and T3 combined. Brick lining shows the average performance on T2 and T4 combined. IoU, Betti error and BwIoU scores were calculated and can be interpreted using figure 7.5. The median score for each α is shown in Table 7.9. The top performing loss function combinations are highlighted in bold for each metric.

IOU: Range (0 - 1) → Higher is better

Measuring pixelwise segmentation performance of masonry joints

BwIOU: Range (0 - 1) → Higher is better

Measuring block instance segmentation performance

Betti: Range (0 - ∞) → Lower is better

Measuring number of topological errors

Figure 7.5: Outline of analysis metrics

Table 7.9: Median performances using increasing proportions of ClDice

ClDice proportion (α)	IoU	BwIoU	Betti
Brick average			
0.1	0.414	0.752	127
0.2	0.425	0.759	112
0.3	**0.427**	**0.776**	74
0.4	0.427	0.764	67
0.5	0.427	0.773	**49**
Stone average			
0.1	**0.514**	0.803	126
0.2	0.500	0.793	114
0.3	0.502	0.811	79
0.4	0.507	**0.813**	69
0.5	0.509	0.812	**34**
Combined average			
0.1	0.464	0.777	156
0.2	0.462	0.776	138
0.3	0.464	**0.793**	94
0.4	0.467	0.788	81
0.5	**0.470**	0.792	**52**

For brick structures, there is a clear improvement in every metric as α is increased from 0.1 to 0.3, while between 0.3 and 0.5, relatively similar scores are achieved for IoU and BwIoU. Stone structures also show a broadly similar trend for BwIoU and Betti number, but not for pixelwise IoU. Given the clear decrease in Betti error with larger values of α, ClDice appears to be improving the topological accuracy of the segmentation. Since this does not lead to large increases in the BwIoU, it is possible that the majority of topological errors occurring with higher soft Dice loss proportions in stone tunnels are isolated false positive segmentations rather than misconnected joints. Removing these has negligible impact on both the pixelwise performance or the connectivity required for joint closure around each block. Larger values of α appear to improve the topological performance without causing notable reductions in the IoU. As a result, to enable a clearer assessment of the impact of ClDice on BwIoU, an α of 0.5 was chosen for further analysis.

7.4.2. Generalisation performance

The BwIoU achieved on each testing dataset for each training regime was examined for different loss functions. However, before the impact of each loss function can be analysed, it is necessary to consider the variation in performance by test Tunnel dataset, as some of the target datasets may be more challenging to segment, leading to a wide spread of BwIoU scores for each loss function. Figure 7.6 shows the average performance by target tunnel. The mean performance for each tunnel is shown in Table 7.10 along with their standard deviations.

Focusing on the ‘Single’ training regime, T1, T2 and T4 achieve similar average performances., although the BwIoU of T3 is significantly higher. However, this difference is either less significant or nonexistent for the other

training methods that involve different structures. Furthermore, T3 performance exhibits the lowest variability of the four tunnels for every training regime. This suggests that T3 has a lower spatial variability in features compared to the other tunnels making it easier for a CNN to perform effectively with only a small volume of targeted training data. Moving from ‘Single’ to ‘Other_st’ and ‘Other_all’ increases the volume of training data. This is shown to improve performance on T1 and T2, suggesting that they have a wider variety of features and so require a larger variety of training data. With the exception of the ‘Single’ training regime, performance on T4 is lower than the other tunnels for all training regimes. This implies that its feature domain has a different shape to the other datasets making it difficult for the trained network to generalise.

A visualisation of the statistical significance of the differences in performance between training regimes is shown in Figure 7.7. For all tunnels, ‘Full_all’ achieves the best performance since it uses the maximum volume of training data. The ‘Other_all’ training regime also achieves excellent performance and is particularly effective on the more homogeneous brick linings. This shows that even with only 3 different training tunnels, a CNN can be trained with powerful generalisation capabilities. The ‘Other_all’ and ‘Full_all’ training methods also show smaller performance variations over the other training methods, highlighting the positive impact of additional training data. This also shows that similar features are present across structures, indicating that with more training tunnels, it may be possible to create a highly generalizable method. The ‘Other_st’ and ‘Full_st’ regimes achieve lower performances for every tunnel compared to their more general equivalents ‘Other_all’ and ‘Full_all’ respectively. This suggests that training should generally be conducted with as much data as possible, regardless of lining type.

Table 7.10: Training regime block connectivity performance comparison

Training Regime	BwIoU (Mean +/- SD)			
	Tunnel 1	Tunnel 2	Tunnel 3	Tunnel 4
Full_all	0.767 +/- 0.017	0.804 +/- 0.018	0.829 +/- 0.017	0.700 +/- 0.025
Full_st	0.760 +/- 0.030	0.782 +/- 0.042	0.853 +/- 0.015	0.671 +/- 0.035
Other_all	0.723 +/- 0.038	0.801 +/- 0.023	0.803 +/- 0.024	0.633 +/- 0.027
Single	0.686 +/- 0.042	0.664 +/- 0.073	0.842 +/- 0.027	0.544 +/- 0.275
Other_st	0.683 +/- 0.053	0.631 +/- 0.32	0.756 +/- 0.036	0.474 +/- 0.074

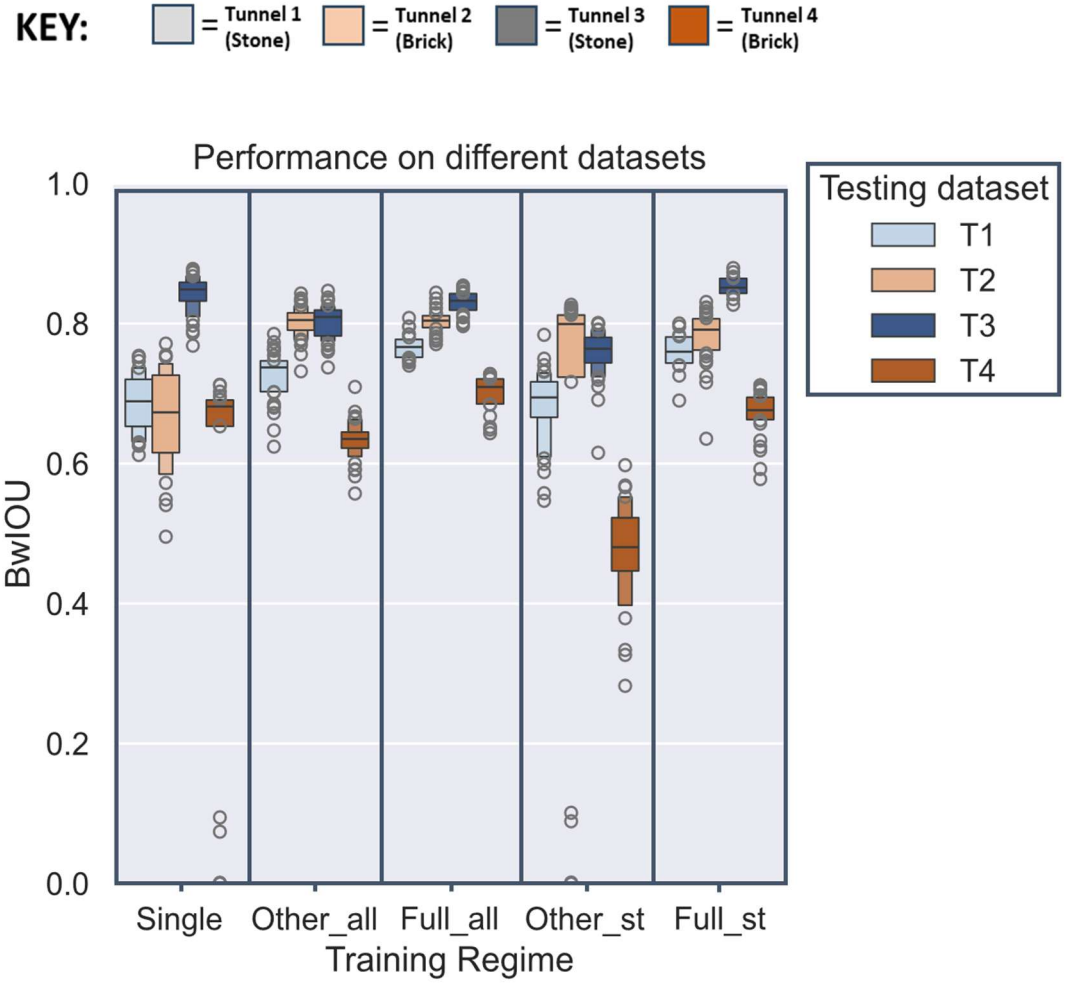


Figure 7.6: BwIoU of networks trained with each training regime plotted by testing dataset. Data from training runs using all loss functions are shown.

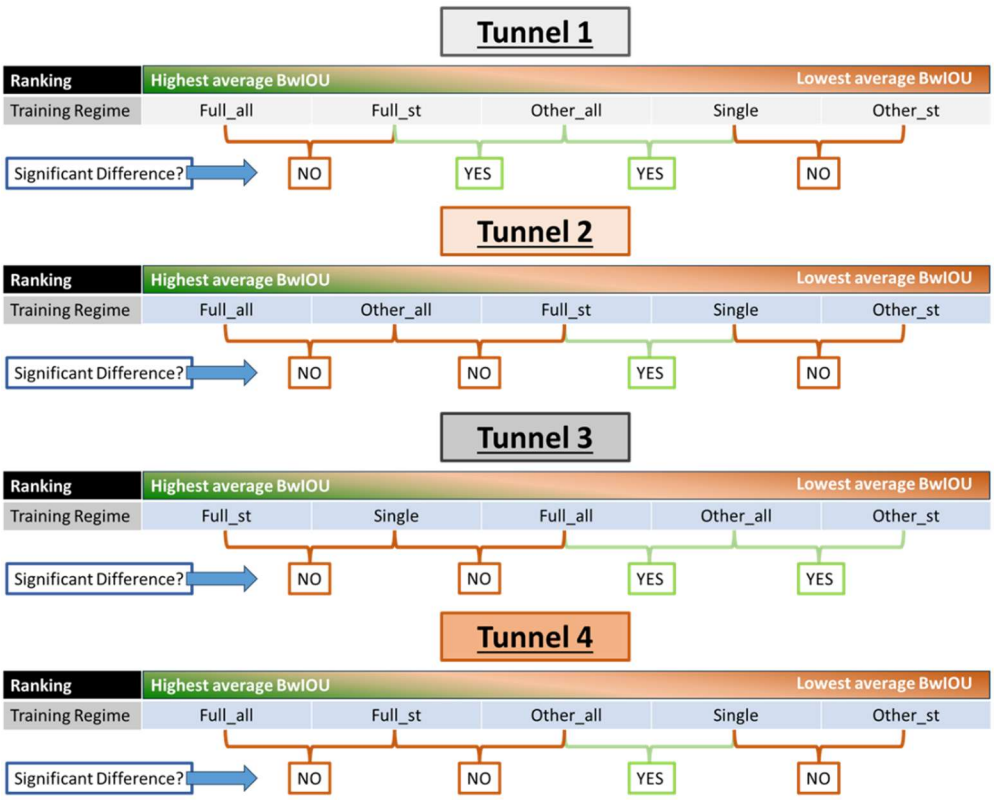


Figure 7.7: Diagram to show the pairwise Tukey HSD comparisons of training regimes for each tunnel. Performance is ordered by mean BwIoU score.

Figure 7.8 shows the variation in performance by training regime and loss function. There is substantial performance variation due to differences in the testing datasets and no significant differences between the loss functions within each training regime. Compared to the pixelwise loss functions, ClDice produces a larger proportion of very low performance occurrences when the neural network becomes stuck in a local minima during training, leading to increased performance variability in the 'Single' and Other_st' training regimes. Nevertheless, the highest BwIoU scores are reached using ClDice with the 'Single' training regime. This suggests that ClDice has the capability to improve performance, but requires larger volumes of training data to be reliable.

The ‘Single’ training regime achieves acceptable performance when Dice or Focal Tversky loss is used for training. Although Focal Tversky achieves on average higher BwIoU scores with lower variability than Dice loss, when training on a single tunnel Dice is recommended, as it would be difficult to reliably tune the Focal Tversky weightings with only limited training data. When a larger volume of training data is used in ‘Other_all’ and ‘Full_all’, ClDice starts to show marginal performance benefits over the pixelwise losses.

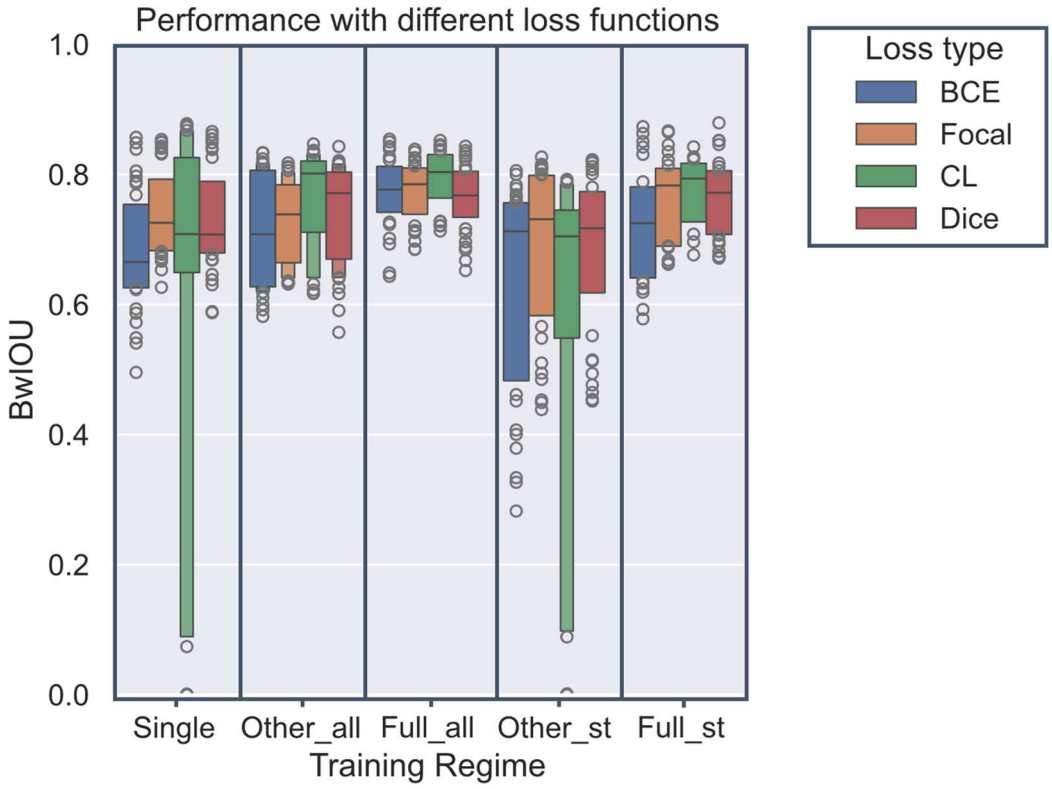


Figure 7.8: BwIoU of networks trained with each training regime and using each loss type. Datapoints from all testing datasets are shown.

7.5. Conclusions

Given a model trained on multiple tunnels for masonry joint segmentation, the following key conclusions are made about performance when applied to a new tunnel:

- Choice of training regime has a much larger impact on performance than the chosen loss function
- Labelling a section of the target tunnel and retraining the model with this section in addition to existing training datasets improves performance over the rest of the target tunnel
- Although similarity between the tunnels used for training and the target tunnel is a key driver of performance, the best performance is achieved in most cases when training with the largest volume and variety of data

Based on these conclusions, the following practical recommendations are made. Training on all available tunnel data consistently achieves higher Blockwise Intersection over Union scores than when trained only on tunnels of similar type. As a result, it is recommended that when other masonry tunnel datasets are available, all datasets are used for training regardless of the tunnel type. Furthermore, if time permits, it is valuable to label a small section of the target tunnel and retrain with all datasets to enable the model to characterise features unique to the target tunnel. In the case where limited analysis time restricts the creation of new training data from the target tunnel, in many cases acceptable performance will be achieved training with only the existing training datasets. However, as shown on Tunnel 4, performance will be poor when a tunnel has features significantly different to those used for training. It is recommended to only skip retraining with target tunnel data

when there is a tunnel with similar lining properties and damages to the target tunnel in the existing training set. If no existing data is available, acceptable performance can be achieved by using a model pretrained on ImageNet and training further using only a small section of the target tunnel. However, a higher variability in performance is expected. Labelling a small testing dataset on the target tunnel to verify the performance of the model is recommended regardless of chosen training regime.

Differences in tunnel features utilised by the neural network are not only caused by changes in lining material or damage level. Domain shift may occur due to differences in the accuracies of the lidar scans when using different scanners or processing methods. These may lead to changes in the local morphology of the lining point cloud. Alternatively, if masonry tunnels from other regions are used, then different lining geometries may also be prominent. In these cases, it is expected that performance would follow a similar trend to that presented here depending on the difference in feature space of the tunnels. Domain shift performance would need to be validated using samples from the different domains. If photographic tunnel data was used instead of 3D lidar data, then the procedure used in this study would need to be replicated to determine the variability of performance on unseen tunnels. Qualitatively, the features most prominent in tunnel lining RGB image data are substantially different to those in a depth map, so it is unlikely that the analysis conducted here could be readily applied to photographic data.

In general, soft Dice loss is proven to be a stable and consistently effective loss function for masonry joint semantic segmentation. Introducing a proportion of ClDice is shown to improve the topological accuracy of the segmentation. However, this can lead to instability during training, resulting

in the loss failing to converge when the training data volume is small. Most importantly, improved topological accuracy brought by ClDice does not result in significantly improved block instance segmentation performance. Further work should be conducted to assess how performance variability can be reduced. The impact of increased training data volume should be compared to increased variety of training data from different tunnels. In addition, topological data analysis is a rapidly growing field and there are multiple other emerging methods for encouraging topologically correct segmentations that could be analysed.

To conclude, this chapter shows that deep learning can be successfully applied for masonry joint segmentation on tunnel linings with small volumes of training data and that with only three tunnels used for training, it is possible to achieve acceptable performance on a previously unseen tunnel. This verifies the ability of the trained masonry joint segmentation models to be applied in practise to the proposed masonry spalling severity segmentation workflow. Finally, it shows that there are clear performance variations depending on the training regime and demonstrates how the applied training regime should be chosen considering both the similarity of lining type and condition between training and testing data, and the total volume of available labelled data.

8. Performance prediction*

**This chapter includes parts of published works (J. M. W. Smith and Paraskevopoulou, 2025)*

We have demonstrated how the masonry damage segmentation workflow can be used on real world structures and assessed how the performance varies between them by studying the performance of CNN based masonry joint semantic segmentation, a key bottleneck in the workflow's performance, in different situations. However, even when we know the optimal conditions for good results, it is still unknown how well the method will perform on a specific structure without extensive manual verification. Given the safety critical nature of tunnel condition assessments, for these methods to be practical in real world applications, engineers need to be able to trust the method's outputs on specific structures. In addition, given the variety of masonry tunnel lining surfaces and the black box nature of deep neural networks, it is difficult to intuitively determine whether a structure's features are in or out of the distribution of the training data and what the requirements are for a specific trained network to perform well. Even for homogeneous concrete structures, concerns about generalisability have limited adoption.

There are multiple methods that can improve neural network performance on unseen structures including data augmentation (Shorten and Khoshgoftaar, 2019), domain adaption methods (Guan and Liu, 2022) and active learning (Budd et al., 2021). However, even when these methods have been adopted, it is vital that performance can be quantified in the target application for the method to be trusted. For the case of masonry joint segmentation, this would typically involve manually labelling a section of tunnel, before assessing segmentation performance against standard quality metrics such as

Intersection over Union (IoU), or Receiver operating characteristic (ROC). This involves time consuming manual annotation by a trained operator. A further issue with this approach is that lining damages and repair work are usually localised, so assessments performed on sample lining data taken at one part of the tunnel may not be representative of the performance throughout the tunnel length.

For deep learning based tunnel analysis workflows to be effectively applied, it is vital that an engineer can quantify the uncertainty in their predictions (Zhang et al., 2021). Ideally, this would involve generating uncertainty maps, so that further analysis can be conducted to verify performance or manually correct a model's segmentation in localised areas. This retains the time saving benefits of these automated workflows where they perform well, but reduces the risk of unverified analysis where there are possibly inaccurate segmentations.

(Feng et al., 2023) investigated how to generate visual explanations of shield tunnel leakage segmentations generated by U-Net style neural networks. They used Grad-CAM++ to explain the reasoning behind the black box U-Net models output leakage segmentations. Grad-Cam uses the derivatives of the feature maps in the last convolutional layer to identify which pixels contribute the most to the segmentation output. However, while this method is useful to investigate which features are leading to incorrect segmentations so can be used to determine a model's limitations, alone it does not indicate how likely the segmentation is correct. Without labelled testing data on the target tunnel, uncertainty quantification methods are required to determine how confident a CNN model is that its segmentation is correct. There have been substantial recent developments in uncertainty quantification for deep learning computer vision applications in the healthcare field (Lambert et al.,

2024). However, there has been limited application of these methods for infrastructure condition assessment tasks.

This section provides an evaluation of which uncertainty quantification methods have the most potential for validating deep learning models applied to real world railway tunnel condition assessment tasks. The chapter compares which methods are most applicable to the task of masonry joint semantic segmentation from lidar data. It considers how uncertainty quantification could be applied in real world masonry tunnel condition assessments and demonstrates the utility of creating uncertainty maps. The study also assesses the correlation between segmentation uncertainty and performance. Overall, this chapter aims to provide guidance on where uncertainty quantification methods could provide value for an engineer, and which would need further development for their insights to be useful and readily interpretable.

8.1. Uncertainty Quantification

As most modern deep learning approaches are black boxes, researchers have looked at various methods for verifying whether a neural network prediction is trustworthy. While many studies have focused on how to explain a model's conclusion (Samek et al., 2021), for the task of semantic segmentation many of these methods have been proven to be either misleading or overcomplex, restricting their interpretability (van Zyl et al., 2024). As a result, this study restricts its analysis to assessing the validity of neural network predictions through uncertainty quantification methods. In the field of deep learning research, uncertainty quantification methods focus on identifying, characterising and quantifying the level of uncertainty in a model's predictions.

8.1.1. Types of uncertainty

Uncertainty can be classified into two types: Aleatoric and Epistemic (Kiureghian and Ditlevsen, 2009). Aleatoric uncertainty refers to inherent randomness in the data that can cause changes in the model predictions. In this application it could be caused by any aspect with unpredictable values for a specific pixel, such as noise in the lidar scan or surface roughness of the masonry lining. Arguably, any feature relationship which is present in the training data, but too complex for the network to characterize, could also be considered as a cause of aleatoric uncertainty as its impacts are effectively random (Valdenegro-Toro and Saromo, 2022).

Epistemic uncertainty represents uncertainty in the output caused by a lack of understanding of the target domain by the model. This is caused by test data having features that are either not represented in the training set or not properly characterised by the model. This results in the test images being out of distribution of the relationships learned by the network leading to unpredictable performance. Features causing epistemic uncertainty will be those most impacted by the domain shift between different tunnels. Examples would be masonry block sizing and shapes, levels and types of damages observed and differences in the mortar joint composition, condition and thickness.

Ideally, the pixelwise SoftMax probability output by the neural network should represent the level of confidence that the model has in its prediction. However, modern convolutional neural network designs are often challenging to properly calibrate within the training data domain and tend to yield confident, but incorrect predictions on out of domain test data (Minderer et al., 2021). As a result, uncertainty quantification methods have been developed to assess a neural networks uncertainty when applied to a specific

dataset. Monte Carlo Dropout (MCD), Test Time Augmentation (TTA) are two commonly used methods that have been chosen for this study.

Unsupervised methods for anomaly segmentation and associated uncertainty estimation have also been developed (Gao et al., 2022). These methods typically involve self-training a student and teacher network. They have been applied to medical images (Adiga V. et al., 2024) and for structural surface damage detection (Lei et al., 2024). However, these methods have not been designed to quantify the aleatoric and epistemic uncertainty of the output from existing trained models, so are not examined further.

8.1.2. Monte Carlo Dropout (MCD)

Monte Carlo Dropout enables the estimation of epistemic uncertainty by varying the neural network design when testing the target dataset. Dropout was initially developed to prevent model overfitting and involves randomly omitting feature detectors during training each time gradients are updated (Hinton et al., 2012). This prevents the learning of overly complex and probably meaningless feature relationships that are unique to each training data sample and so worsen performance on test data.

While the network is usually frozen and all neurons are retained for testing, MCD involves applying dropout at test time. The testing data is run through the network with different neurons dropped out each time and the resulting variations in output give an indication of the level of uncertainty in the network's predictions. First proposed by (Kendall et al., 2017) and theoretically proven as a Bayesian equivalent in (Gal and Ghahramani, 2016), test time dropout approximates the posterior distribution of the network's weights, by Monte Carlo sampling the network's predictions. By

assessing the model variance on a particular test image, it is possible to assess the epistemic uncertainty of the prediction (Kendall and Gal, n.d.).

MCD has been applied relatively extensively for semantic segmentation tasks in the medical field, where quantifying uncertainty levels on medical imaging tasks are vital for a clinician to make informed decisions about a patients' treatment options (Arega et al., 2021; Laves et al., 2019; Nair et al., 2020; Seeböck et al., 2019; Wickstrøm et al., 2020). However, while some studies have applied MCD for uncertainty quantification of construction object segmentation (Vassilev et al., 2024) and concrete damage assessment (Dos Santos et al., 2024; Rathnakumar et al., 2023; Sajedi and Liang, 2020), it has not been applied to semantic segmentation tasks on older less homogeneous infrastructure such as masonry lined tunnels. This is despite the need for similar safety critical decisions to be made on those structures based on a neural network segmentation output.

8.1.3. Test Time Augmentation

Test Time Augmentation (TTA) is a commonly applied method for estimating the heteroscedastic aleatoric uncertainty in a model's prediction. While data augmentation during training to improve test time performance is well documented (Shorten and Khoshgoftaar, 2019), (Ayhan and Berens, n.d.) were the first to apply test time augmentation to help understand the aleatoric uncertainty in their model's output. They made 128 copies of their test samples and ran them through standard training image augmentations to produce 128 variants of their test data. They then put these images into their trained image classification network and observed the variations in outputs between the transformed images. (G. Wang et al., 2019) later formalised their method and assessed its potential for uncertainty quantification using

the Volume Variation Coefficient (VVC) of segmented structures for brain tumor segmentation. They showed a negative correlation between the VVC, calculated by dividing the segmentation volume variance by it's mean, and the segmentation dice score. Although less commonly adopted than MCD, TTA has been adopted for uncertainty quantification of other types of medical images (Lin et al., 2023; Zheng et al., 2022). It has not been applied to structural condition assessments.

8.2. Purpose of Chapter

The object of this chapter is to analyse the potential of MCD and TTA for quantifying the uncertainty of neural networks trained for masonry joint segmentation from lidar data. This will increase the trustworthiness of automated masonry lined tunnel condition assessment procedures that rely on deep learning based masonry joint segmentation. The study aims to provide a method for automatically highlighting anomalous segmentations to an engineer so that they can be manually adjusted or removed. It provides the following three contributions:

- A comparison of MCD with TTA for structural condition assessment uncertainty quantification.
- An analysis on whether uncertainty can effectively predict performance.
- Consideration of the usefulness of generated uncertainty maps.

The study adapts the U-Net style neural network outlined in Chapter 6 to analyse two uncertainty quantification methods:

- Monte Carlo Dropout (MCD)
- Test Time Augmentation (TTA)

The study compares these methods and investigates the correspondence between uncertainty and model performance.

8.3. Methods

After labelling, the images were upsampled such that each tunnel had the same average number of pixels per masonry block. Tunnel 3 has the largest masonry blocks, so 1,2 and 4 were adjusted to match. Data from each tunnel was then split into training, validation and testing sets using the full_all format applied in Chapter 7 and a 4:1 validation to training data split. Finally, images were split into 384x384 pixel patches. This was to ensure that the neural networks could be trained within the available VRAM constraints.

8.3.1. Neural Network Training

In order to focus on uncertainty quantification performance, a basic U-Net style neural network was chosen for the analysis. The network architecture from (Do et al., 2020) was used. This consisted of 4 down sampling convolutional layers in an encoder followed by 4 up convolutional layers in the decoder. The model was trained using the hyperparameters outlined in Table 8.1 on the AIRE workstation outlined in Chapter 4.

Table 8.1. Chosen neural network architecture and hyperparameters

Architecture	Optimiser	Weight decay	Batch size	Initial learning rate	Pretraining
U-Net	AdamW	0.002	8	0.001	ImageNet

8.3.2. Quantifying Epistemic and Aleatoric uncertainty

It is necessary to wholistically assess the performance of each uncertainty quantification method given the wide possible variations in feature distribution of both the trained neural network selected and target tunnel lining. As the aim of the uncertainty quantification is to indicate the neural network's performance and applicability to a specific tunnel, performance on both in and out of distribution tunnel data needs to be assessed.

Four different tunnels were available for this study, so the neural network was trained three times using different hyperparameters to simulate a total of 12 different domain-shift scenarios. This creates different levels of epistemic uncertainty. Details of these networks are described in Table 8.2. The differences between the trained networks act as a proxy for the wide variety of possible differences in features between training and testing data tunnels due to their geometries, material type and damage levels. Aleatoric uncertainty also needs to be modelled. In order to artificially create uncertainty in the test data, random Gaussian and Perlin noise was added to each test image. Two levels of noise were chosen, with scale factors of 0.15 and 0.3 applied to the magnitude of the noise. All data augmentations were implemented using the Albumentations Python library.

Networks A, B and C were trained with random vertical and horizontal flips. Through trial and error, the data augmentations that lead to the best test data performance were determined and applied for Network A. Network A was trained with the full set of data augmentations described within section 6.6.4.

Table 8.2: Differences between trained neural networks

Network	Description	Augmentations	Training epochs
A: Well generalised	This is a well trained network that generalizes effectively to operate on different structures. Due to data augmentations, a wider feature distribution is represented in the networks training.	High	1200
B: Overfitted	This network has been trained to achieve excellent performance on the training set. It has a narrower feature distribution, so features from other tunnels are more likely to be poorly represented in the network.	Low	1000
C: Underfitted	The network has been undertrained so does not learn a detailed representation of the target domain. However, it retains a wide feature distribution from pretraining.	Low	200

8.3.3. Neural Network Performance

Each network was trained on Tunnel 1 before being applied to the test data of all 4 tunnels. Performance was assessed using the BWIoU outlined in section 7.1.3.

As MCD requires training with dropout enabled, the networks were trained with and without dropout in order to assess if including dropout has negative performance impacts. Performances of the networks trained without dropout are shown in Table 8.3 and the outputs are visualised in Figure 8.1. It is clear that Network A has better generalisation performance than B and C. The overfitting of Network B to Tunnel 1 yields the best results on Tunnel 1 test data at the expense of Tunnels 2, 3 and 4 performances. Performance on Tunnels 2 and 4 is worse than Tunnel 3. This is likely due to the differences in features between stone and brick lined tunnels. Although the decrease in performance when moving from Network A to B and C is qualitatively visible

in Figure 8.1 for tunnels 2 and 4, the IoU shown in Table 8.3 decreases substantially. This is because even when the joints are segmented largely correctly, small gaps in the joints connect adjacent block instances, leading to a substantial breakdown in block segmentation performance.

Table 8.3: Neural network performance without dropout

Network	BWIoU on test data			
	Tunnel 1	Tunnel 2	Tunnel 3	Tunnel 4
A	0.62	0.38	0.68	0.33
B	0.63	0.02	0.44	0.02
C	0.60	0.07	0.71	0.09

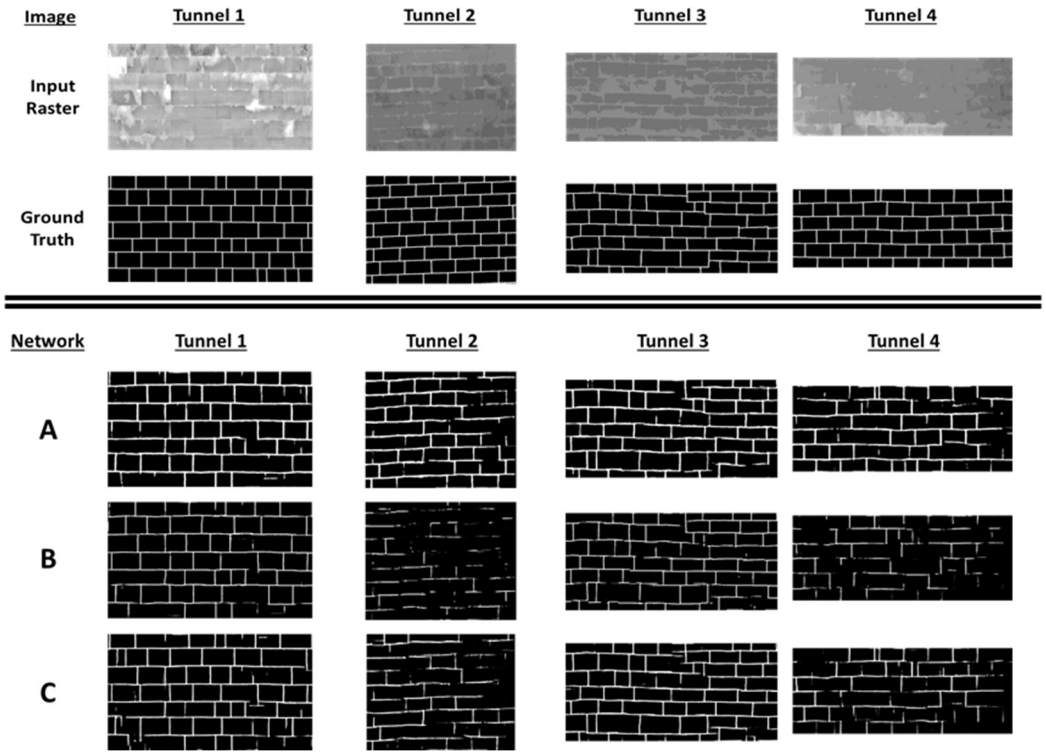


Figure 8.1: Comparison of neural network segmentation outputs on a section of testing data from each tunnel.

The model was then trained with dropout enabled, with a dropout probability of 0.5 for each neuron. As shown in Table 8.4, this produces slightly higher, but largely similar results. Tunnel 4 performance is particularly improved. This is possibly due to the regularising effect of dropout.

The impact of the artificially added noise on neural network performance was also assessed. Figure 8.2 shows how added noise impacts the segmentations of a section of testing data taken from Tunnel 3. Figure 8.3 visualises the distribution of performance decreases for each tunnel segment when the amount of added noise is increased. It can clearly be seen that for every network, adding noise decreases performance. For Network B, the change is less pronounced. This is likely because it achieves poor performance on Tunnels 2,3 and 4 in the no added noise case, but good performance on Tunnel 1. Adding noise has little impact when the performance is already low. This is reflected in the visualizations in Figure 8.2.

Table 8.4: Network performance with dropout. Dropout probability = 0.5

Network	BWIoU on test data			
	Tunnel 1	**Tunnel 2**	**Tunnel 3**	**Tunnel 4**
A	0.65	0.42	0.67	0.46
B	0.67	0.02	0.56	0.19
C	0.69	0.2	0.71	0.19

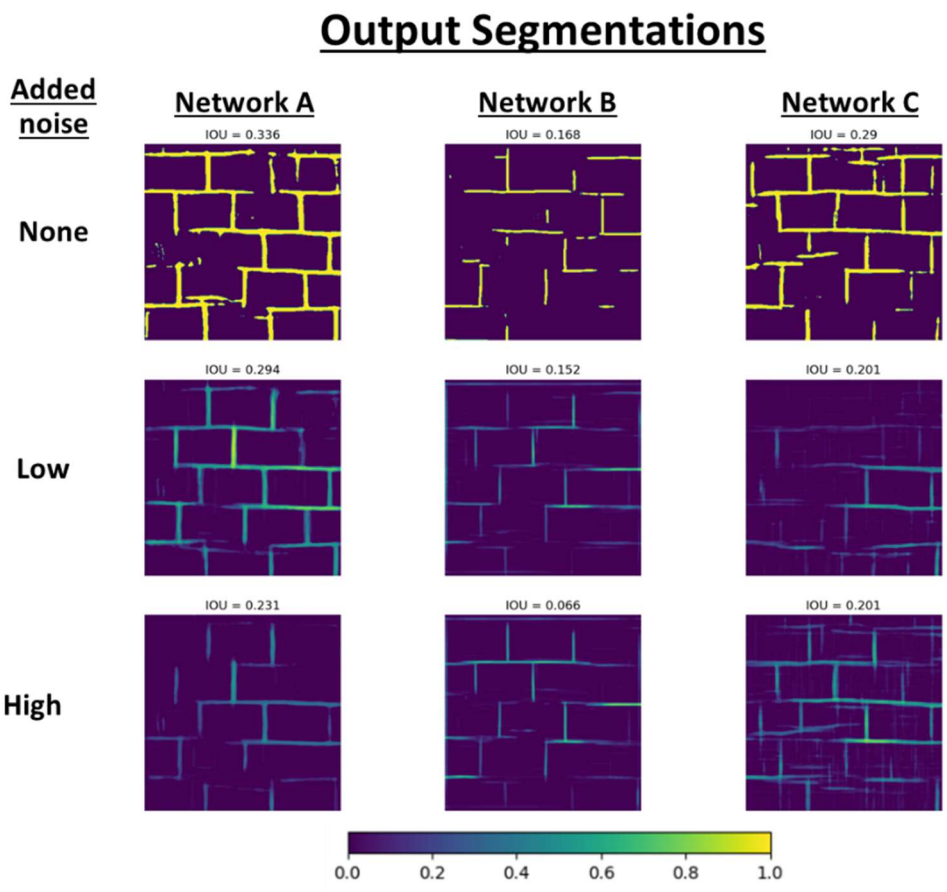


Figure 8.2: Comparison of segmentation sigmoid output by neural network computed on a single patch of testing data given different levels of added noise in the input image patch. The outputs for Low and High noise levels are the average of the output over 5 different added noise maps.

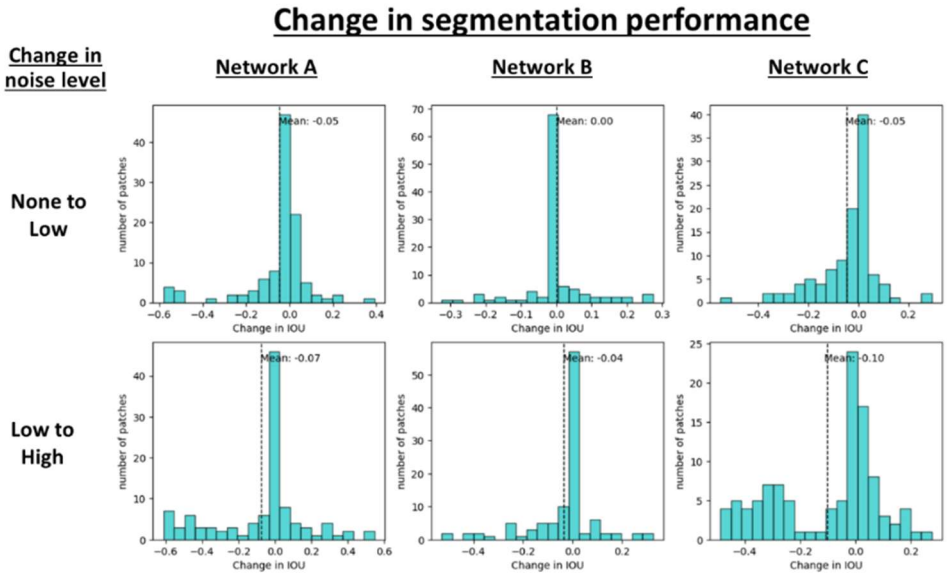


Figure 8.3: Histograms showing the change in IoU for each patch when the magnitude of the added noise is increased. The mean increase in IoU over all patches is shown at the top of each figure.

8.4. Uncertainty metrics

Both MCD and TTA produce a number of output segmentations that need to be compared for uncertainty to be quantified. Two different uncertainty metrics were chosen.

- Uncertainty Mean Intersection over Union (UMIoU). The IoU between each output segmentation and the mean of all generated segmentations is calculated leading to an IoU score for every Monte Carlo sample. The UMIoU is the mean of all the sample IoUs. Higher uncertainty leads to a lower UMIoU.
- Area Variation Coefficient (AVC). This is a 2D version of the VVC used within (G. Wang et al., 2019). A summation of the number of predicted masonry joint pixels for each Monte Carlo sample is performed. The AVC is calculated by dividing the standard deviation

between these summations by their mean. Higher uncertainty leads to increases in the AVC.

Uncertainty maps were generated for each segmented image patch by calculating the standard deviation between the segmentation prediction of each Monte Carlo sample for each pixel. Each pixel value on the uncertainty map is then set as the calculated standard deviation.

8.5. Test Time Augmentation

Test time augmentation was applied to each of the three neural networks once trained with dropout enabled. Gaussian noise, Perlin noise, brightness and contrast shifts, and random vertical and horizontal flips were chosen as the augmentations. Each 384*384 test image crop was augmented in 50 different ways. The UMIoU and AVC were calculated between each of the 50 segmentations for each image.

The generated uncertainty maps are visualised in Figure 8.4 for an image patch with varying levels of added noise in the input image. For Network A, Although the AVC increases with increasing noise levels, visually, there appears to be fewer areas of uncertainty. Inspecting Figure 8.4, the network does not identify as many joint locations with the increase in noise. This suggests that if the noise level is significant enough to completely obscure image features, then the network will be more confident in its prediction even though it is incorrect. The AVC is able to reflect the increased uncertainty as it is normalised by the area of predicted joints, so is robust to the decreased segmentation area. Networks B and C show decreases in the UMIoU with increasing noise, it is unclear that the level of uncertainty is increasing from viewing the uncertainty maps alone.

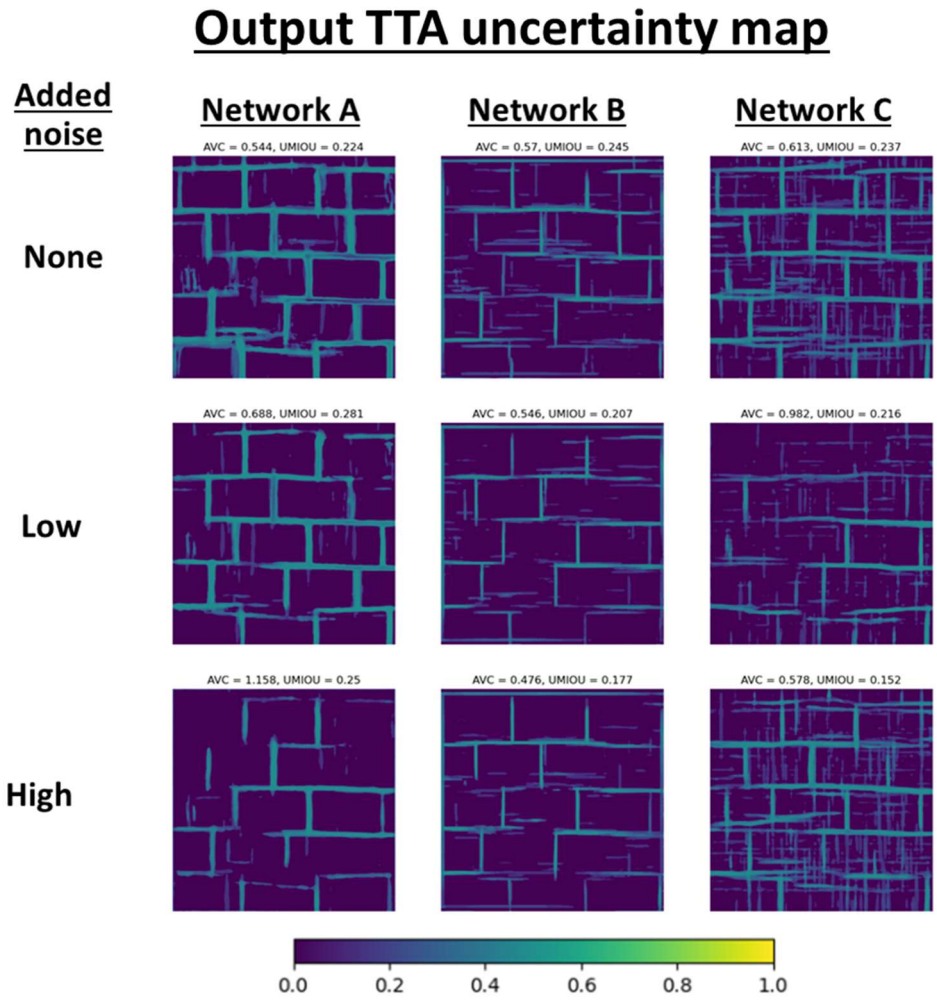


Figure 8.4: TTA uncertainty maps for a patch of Tunnel 3 data given different trained models and synthetically added noise magnitudes. The standard deviation of the sigmoid output over all samples is shown for each image pixel (larger is more uncertainty).

The distribution of increases in AVC over all image test patches is visualised within the histograms in Figure 8.5. The mean AVC value increases for all networks when the level of added noise is increased. For Network B, the increases are lower. As Network B has poor generalization performance, it is unable to effectively characterize out of distribution data. It is less able to reduce its prediction confidence in uncertain situations. As a result, it likely

produces universally more confident predictions due to incorrect identification of features in the test data as those from the training data.

The AVC and UMIoU uncertainties on each patch are plotted against segmentation performance in Figure 8.6. There are no clear relationships between uncertainty value and segmentation performance observed for TTA.

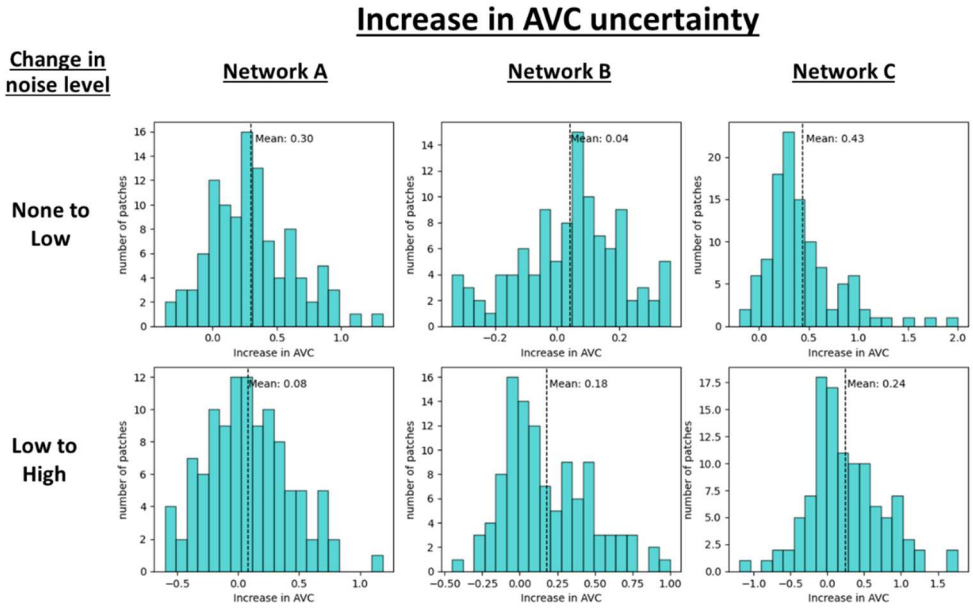


Figure 8.5: Histograms showing the change in AVC uncertainty for each patch when the magnitude of the added noise is increased. The mean increase in AVC uncertainty over all patches is shown at the top of each figure.

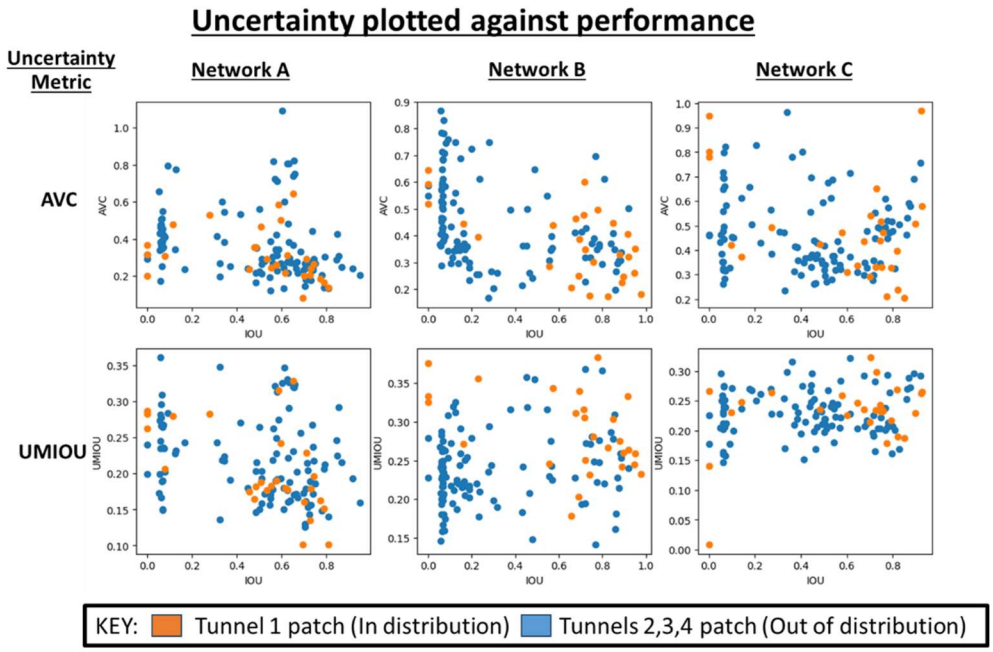


Figure 8.6: TTA uncertainty values plotted against segmentation performance for each trained neural network. AVC and UMIoU uncertainty metrics are compared.

8.6. Monte Carlo Dropout

Monte Carlo dropout was analysed for each of the three trained neural networks. Dropout was applied during inference with a probability of 0.5 per layer. Each 384*384 test image crop was put through the trained network 100 times. The UMIoU and AVC were calculated between each of the 100 Monte Carlo sampled segmentations for each image. The generated uncertainty maps are visualised in Figure 8.7 for an image patch from both a section of Tunnel 1 data, which should have a low level of epistemic uncertainty, and a section of Tunnel 3, which should have a higher level of epistemic uncertainty.

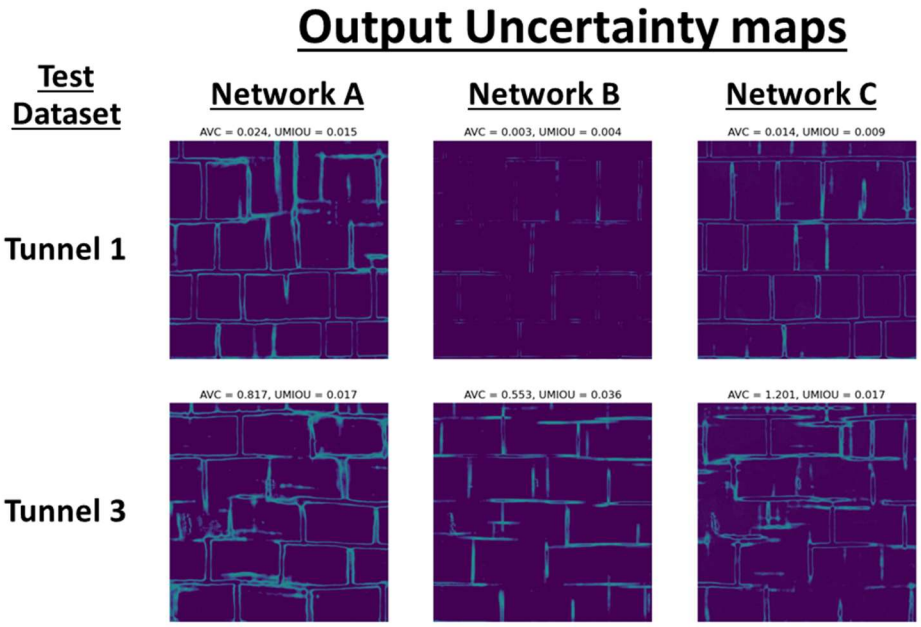


Figure 8.7: MCD uncertainty maps for a patch of Tunnel 1 and Tunnel 3 data given different models. The standard deviation of the sigmoid output over all samples is shown for each image pixel (larger is more uncertainty).

For Network A, the uncertainty maps clearly show the locations where varying segmentations are possible. For example, there is clearly uncertainty in Network A's segmentation at the top of the Tunnel 1 patch. Observing the regularity of the masonry, it is likely that at this location masonry cracks are being predicted as joints. For Network B, there is a substantial increase in both the AVC and the level of uncertainty that can be qualitatively observed. The change in epistemic uncertainty is most extreme for Network B as it is overfitted to Tunnel 1. This demonstrates how uncertainty maps can be used to determine which tunnel areas are likely to be within the distribution of the trained model.

The correlation between MCD uncertainty and segmentation performance can be observed in Figure 8.8. There is a weak negative correlation between AVC and IoU for Network B and C, however it is less pronounced for Network

A. This is possibly because Network A generalizes better to unseen data than Network B and C. This would cause epistemic uncertainty to have less of a negative performance impact than for Networks B and C. An alternate explanation is that Network A may capture a broader variety of feature relationships than Networks B and C, due its exposure to a wider variety of training data. The more complex relationships are more adversely impacted by dropout than the simpler relationships in Networks B and C. This would lead to MCD generating universally higher levels of uncertainty and suggests that MCD can be used to indicate the generalizability of a trained network. A further analysis covering a wider variety of datasets would need to be conducted to fully separate the impacts of these factors.

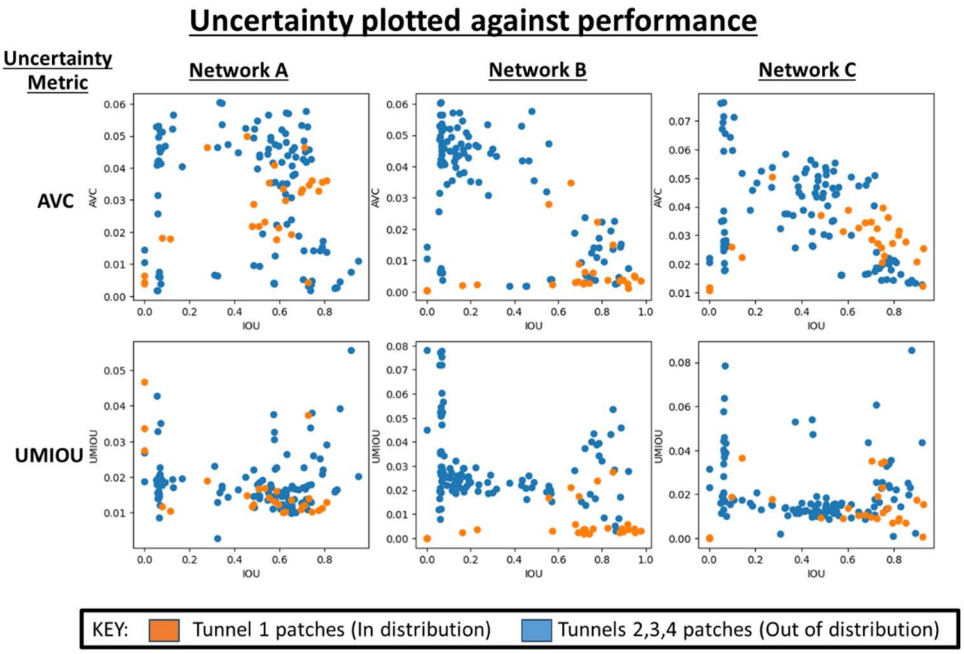


Figure 8.8: MCD uncertainty values plotted against segmentation performance for each trained neural network. AVC and UMIoU uncertainty metrics are compared.

8.7. Uncertainty visualisations

Having analysed the capabilities of each uncertainty quantification method, it is important to be able to visualize each metric in an accessible and informative way. We propose projecting the AVC scores of each 384*384 image patch onto the output segmentation maps. This enables an engineer to efficiently scan the tunnel lining for areas with high uncertainty. The pixelwise uncertainty maps should be inspected when a specific area requires a more detailed inspection. They can then be used to view alternate possible segmentations. An example of this is shown applied to a section of the Tunnel 3 test data in Figure 8.9. The correlation between segmentation performance and MCD uncertainty can be observed.

Without uncertainty maps, it would be necessary for an engineer to inspect every segmentation map in detail to validate the segmentations. However, Sections 3.3 and 3.4 show that although high levels of uncertainty are correlated with poor performance, it is possible for a patch with low epistemic or aleatoric uncertainty to also generate a poor segmentation. As a result, areas with poor performance, where the segmentation needs to be manually analyzed and corrected cannot be exclusively determined using uncertainty values. It is necessary for an engineer to take a holistic approach when identifying locations with poor segmentation performance. The following workflow is suggested:

1. An engineer should identify the typical masonry block dimensions from the segmentation maps. If there are multiple types of masonry present, then the engineer should conduct the following steps over each type of masonry in turn as uncertainty values are not directly comparable between areas with substantially different properties. TTA and MCD image patches should be sorted by uncertainty level.

2. Starting from patches with the highest uncertainty, the patch predictions should be observed alongside the pixelwise uncertainty values and the input depth map. If the predicted joint locations do not appear realistic, then the segmentation should be manually corrected. The MCD pixel uncertainty maps show segmentation outputs with variants of the trained neural network. They can therefore be used as a guide to identify more realistic segmentation candidates.
3. Step 2 should be conducted for patches with progressively smaller uncertainties until the observed patches have qualitatively acceptable segmentations.
4. It is necessary to account for areas where there may be poor segmentation performance despite a low level of uncertainty being identified. These regions are likely caused by abnormalities in the input depth map caused by tunnel features that have not been encountered during training and are challenging to accurately identify from the depth map alone. While many of these cases will lead to epistemic uncertainty, it is possible for the network to be confidently incorrect if a joint is not visible in the depth map. This may occur, for example, if the mortar is level with the masonry surface. In addition, high levels of noise are not always detected by TTA as aleatoric uncertainty if no reasonable segmentations can be generated. As a result, the engineer should conduct further pixelwise segmentation verification in areas they have identified as anomalous during their on-site qualitative inspection of the tunnel.

Although this method is not guaranteed to remove all incorrect segmentations, it is a cost effective procedure for improving segmentation performance given limited available manual analysis time and would substantially reduce the analysis time compared to full manual labelling of masonry block locations.

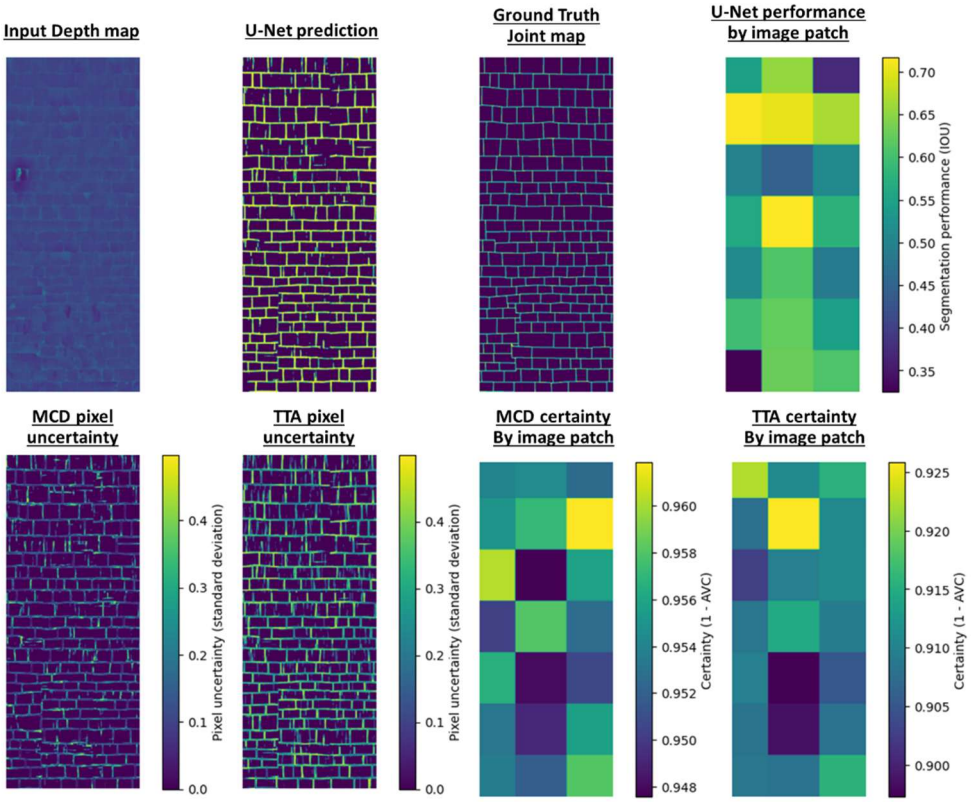


Figure 8.9: Patchwise and pixelwise uncertainty metrics visualised over an area of Tunnel 3 to show how they can be interpreted over a larger area of tunnel. Pixel uncertainty shows the standard deviation of each pixel (Larger is more uncertainty). Patch certainty is 1 minus the AVC uncertainty for each input image patch (Larger is less uncertainty).

8.8. Discussion and Conclusion

Both Test Time Augmentation (TTA) and Monte Carlo Dropout (MCD) generate uncertainty maps that can aid in the interpretability of deep learning based masonry joint segmentations. MCD enables alternate possible segmentations to be viewed, visualizing the sensitivity of the output to the neural network training environment. The Area Variation Coefficient (AVC) can be used to demonstrate how increased epistemic uncertainty leads to decreased segmentation performance. However, the uncertainty values generated by Monte Carlo dropout are sensitive to aleatoric uncertainty. Adding noise decreases the AVC of MCD uncertainties.

TTA shows how small changes in the input data lead to changes in the segmentation output. Qualitatively, the produced TTA uncertainty maps are strongly dependent on how robust a trained network is to noise. The AVC of TTA outputs is increased with increased noise levels, enabling it to be used as an indicator of the quality of the input images and the resulting aleatoric uncertainty. However, it is not shown to correlate with segmentation performance, as this is strongly driven by epistemic uncertainty caused by domain shift between the tunnels analysed.

For both TTA and MCD, the UMIoU is not shown to be a useful metric of uncertainty. The UMIoU does not correlate in most cases with the AVC or the perceived level of uncertainty observed in the uncertainty maps. As a result, the AVC is proposed as the standard segmentation uncertainty evaluation metric.

There is a substantial runtime increase when implementing uncertainty quantification methods. The runtime increase is proportional to the number of augmentations or dropout variations that are assessed, since inference

needs to be computed for every Monte Carlo and augmentation sample. For this study, implementing TTA and MCD increased the inference time by approximately 1500%, as 100 MCD samples and 50 TTA samples were used. Reducing the number of samples reduces the effectiveness of the method as it is necessary to generate a distribution of outputs in order to more confidently determine the mean and standard deviation of them. With a low number of Monte Carlo samples, it is possible that key augmentation/ dropout permutations are missed, generating misleading results. It is recommended that MCD and TTA should only be implemented on standard office hardware when the computational cost of inference is small. Alternatively, cloud compute instances could be rented for the inference process. This would prevent the analysis time forming a bottleneck in conducting a condition assessment without requiring the purchase of expensive specialist hardware that may only have occasional use over the lifetime of a project.

To conclude, both TTA and MCD provide valuable insights into the uncertainty of masonry block segmentation outputs and AVC score maps can be effectively used to indicate to a tunnel inspector the locations where a neural network has high levels of uncertainty. Within a specific tunnel, the AVC value correlates with segmentation performance, enabling an inspector to easily identify the most effective locations to conduct manual validation or correction of the output masonry joint segmentation map. However, there are limitations with using uncertainty alone as a proxy for segmentation performance since their power is dependent on the specific trained neural network. It is suggested that the level of uncertainty is calibrated against segmentation performance on samples of unseen testing data before being applied in practice. A well trained and generalised neural network should

generate more strongly correlated uncertainty scores. However, it is still necessary to conduct a qualitative visual inspection of a tunnel lining to identify where obvious lining anomalies may impact segmentation performance.

In order to determine how uncertainty maps could be integrated into real world tunnel condition assessments, further work needs to be conducted to analyze how accessible and interpretable these methods are for engineers who are not familiar with machine learning. Furthermore, asset managers need to be consulted to determine how far uncertainty quantification methods would improve their perception of the trustworthiness of automated tunnel analysis workflows.

9. Workflow performance assessment*

**This chapter includes parts of published works: (Smith et al., 2024)*

With the theory behind the proposed masonry tunnel spalling severity segmentation workflow verified in Chapter 5 and the performance of the joint and damage segmentation steps of the workflow assessed on real tunnels in Chapters 6 and 7, it is necessary to evaluate the full workflow in the context of real world tunnel condition assessments. The aim of this chapter is to assess the overall performance of the workflow proposed in Chapter 5 on real tunnel data to determine its limitations and its place in the future of masonry lined tunnel condition assessments. It then outlines recommendations for future work that would be required before the workflow could be reliably applied by industry.

The workflow was evaluated in this chapter on sections of the stone lined tunnel T1 and brick lined tunnel T2 alongside unseen tunnel T5. The workflow was run through 10m long sections of each tunnel on device L2 using the U-Net Mit_b5 model evaluated in section 6.7 for masonry joint segmentation and the UNet++ MobileNet_v2 model proposed in section 6.8 for masonry damage segmentation. Each CNN was trained on sections of tunnels T1, T2, T3 and T4 previously used for training in section 7.3.5. A quantitative performance analysis was first conducted on separate test sections of T1 and T2, which have existing ground truth spalling severity maps. T1 was selected as a representative stone lined tunnel and T2 as a brick lined one. T1 performance was then used to investigate the sensitivity of the workflow output to the Damage Proportion Threshold (DPT) used in the plane fitting stage. Afterwards, a qualitative evaluation was conducted on

a section of Tunnel T5 against a manually recorded visual inspection report. Using T5 as a case study, three sets of analysis were conducted:

1. **Run time analysis:** The processing time of each section of the workflow was evaluated with mobile netv2 and mit_b5 encoders. The impact of point cloud density on processing time was explored.
2. **Point cloud density investigation**. The impact of point cloud density on processing time and joint segmentation was explored.
3. **Qualitative real-world performance analysis**. The output of the automated workflow was compared with that of a manually identified damage map and change detection outputs.

9.1. Quantitative analysis on T1 and T2

The full spalling severity segmentation workflow was evaluated on test sections of T1 and T2 subsampled to 4mm point spacing. With the manually defined damage depth map set as the ground truth, Table 9.1 shows the IoU of the automated workflow for each level of spalling severity segmented. The longer evaluation time on T2 was due to the smaller sized, but increased number of blocks on the brick lining. As conducted in Chapter 7, the depth map rasters of the smaller blocks were resized such that average block dimensions matched that of the larger stone blocks in T1. Operating on larger images lengthened the runtime of each stage of the workflow. The sections of T1 and T2 used for testing are shown in Figure 9.1. Evaluation was conducted with no manual input, using default parameters. Figure 9.2 shows the output of the workflow on T1. As performance was evaluated on the damage locations reprojected onto the point cloud, the IoU was evaluated per point.

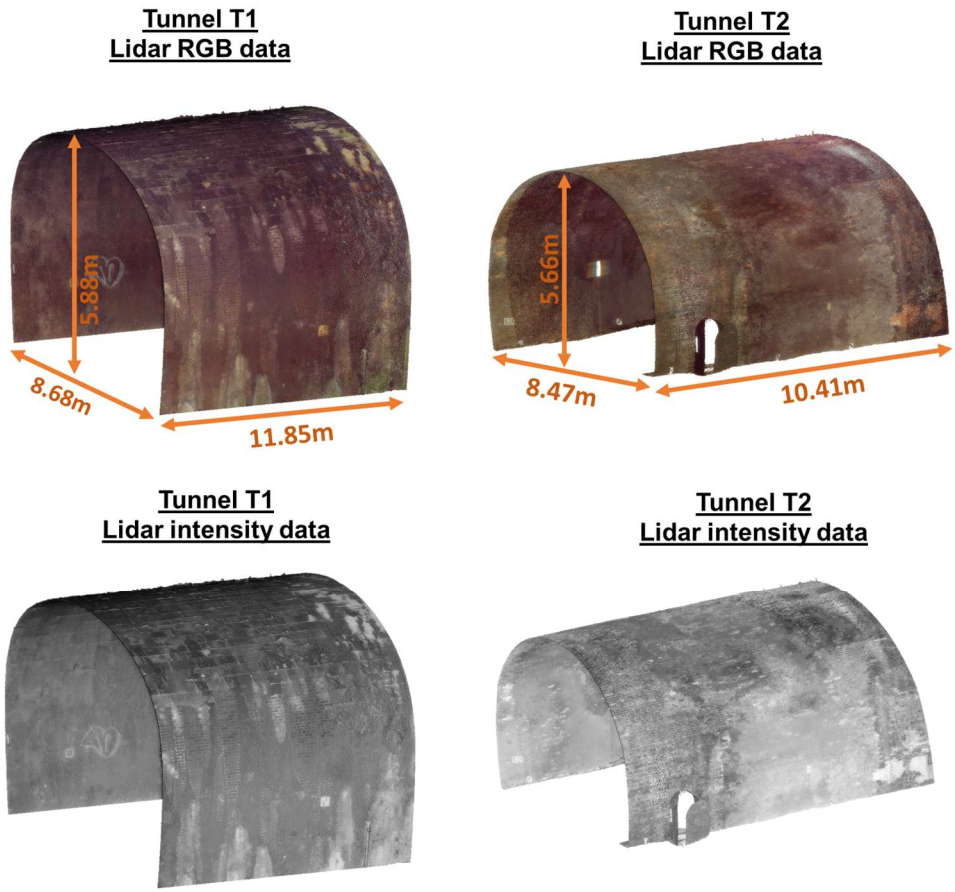


Figure 9.1: Visualisations of the sections of tunnels T1 and T2 used for testing the trained spalling severity segmentation workflow. Point cloud RGB and lidar intensity data recorded during the point cloud surveys of each tunnel are shown.

Table 9.1: Performance of the trained workflow on the test sections of tunnels T1 and T2. Performance is shown by severity level using the manual method's output as the ground truth.

Tunnel	Number of blocks	Workflow runtime (s)	>S1 damage IoU	>S2 damage IoU	>S3 damage IoU
T1	755	180	0.648	0.507	0.496
T2	7101	613	N/A	0.237	0.432

9.1.1. Performance discussion

As T1 is stone lined but T2 is brick lined, the spalling severity levels for each tunnel are defined differently in Network Rail's standards (Network Rail, 2016). The depth thresholds for each severity level are shown in Figure 9.2.

For T1, S1 performance was greater than S2 and S3, although all IoU scores are acceptable. Performance on T2 was worse than T1. For S3 spalling, this was 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 small amount of S2 and S1 spalling within the T2 test data yields the S1 and S2 results for T2 inconclusive.

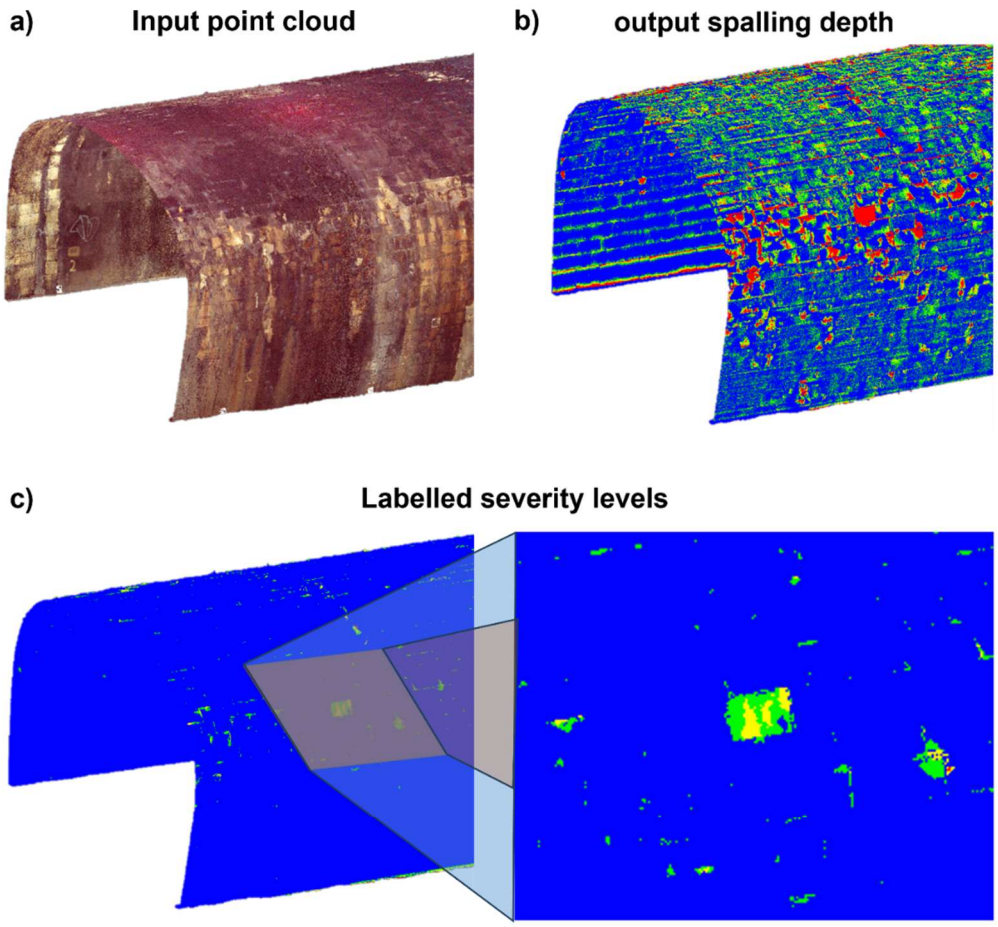
For both tunnels, S3 spalling was substantially more extensive than S1 and S2. It was also more difficult to accurately manually segment S3 areas, as the small 10mm threshold makes it challenging to identify against noise. For the automated workflow, to achieve a similar level of S3 segmentation performance, the damage segmentation neural network performance needs to 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 was mostly caused by the spatial extent of spalling fully covering multiple blocks. This led 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 9.2 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 was greater for both tunnels than the pixelwise segmentation performance and the workflow achieved adequate S3 localisation accuracy against that typically required for both T1 and T2. For all severity levels, T1 produced a more conservative segmentation, with more false negatives, than false positives. The reverse was true for T2.

Table 9.2: 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 (>100mm)	**T2 (>50mm)**	**T1 (>40mm)**	**T2 (>20mm)**	**T1 (>10mm)**	**T2 (>10mm)**
-18.0%	N/A	-37.3%	91.5%	-31.4%	-7.98%



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

Figure 9.2: Progression of the overall workflow on tunnel T1 (a) Input point cloud (RGB colour data shown) (b) Output of workflow showing defect depth values (c) Spalling severity output (key in table)

9.1.2. DPT value selection

The Damage Proportion Threshold (DPT) is an important parameter that must be optimised to maximise workflow performance. The DPT determines the maximum proportion of pixels within a block that when identified as damaged, leave an acceptable number of undamaged pixels to reliably fit an undamaged surface to the block. If the DPT is exceeded, then the block is discarded, and the undamaged block surface location is interpolated from neighbouring blocks. 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 9.3. 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 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.

Table 9.3: Pointwise IoU score for different spalling severity categories in T1 and T2. Results when different DPT values are selected are shown.

Plane fitting Damage Proportion Threshold	IoU for spalling greater than S1		IoU for spalling greater than S2		IoU for spalling greater than S3	
	T1 (>100mm)	T2 (>50mm)	T1 (>40mm)	T2 (>20mm)	T1 (>10mm)	T2 (>10mm)
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

9.1.3. Sensitivity analysis

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. Figure 9.3 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.

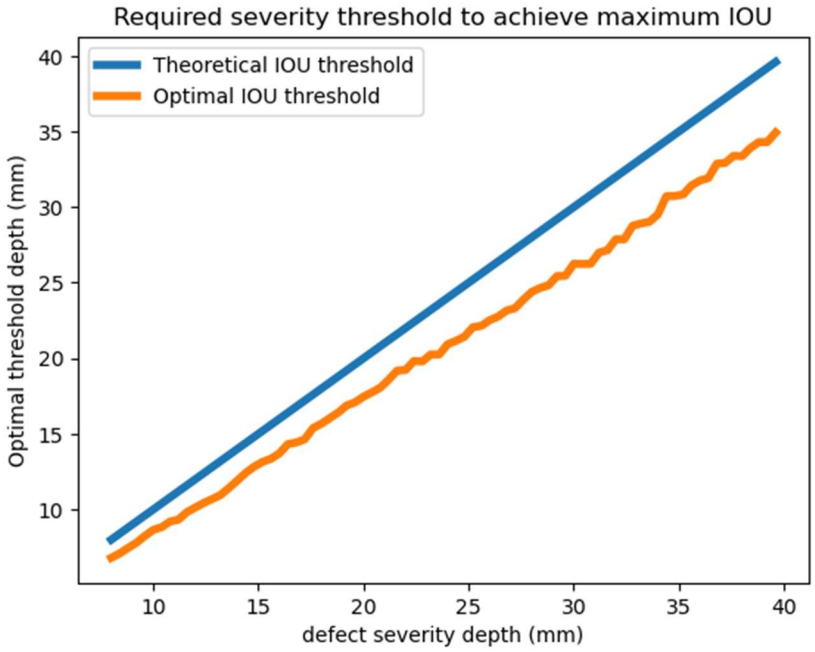


Figure 9.3: Defect depth threshold that generates a segmentation with the highest IoU score for each target level of defect depth within T1.

9.1.4. Local performance visualization

An in-depth comparison of workflow performance on two areas of tunnel is visualised in figures 9.4 and 9.5. The output of the workflow on the entire testing sections of T1 and T2 are provided as supplementary data. The area of tunnel in Figure 9.4 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.

Figure 9.5 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 the manual method, the algorithmic nature of the automated method ensures that the results are repeatable and understandable.

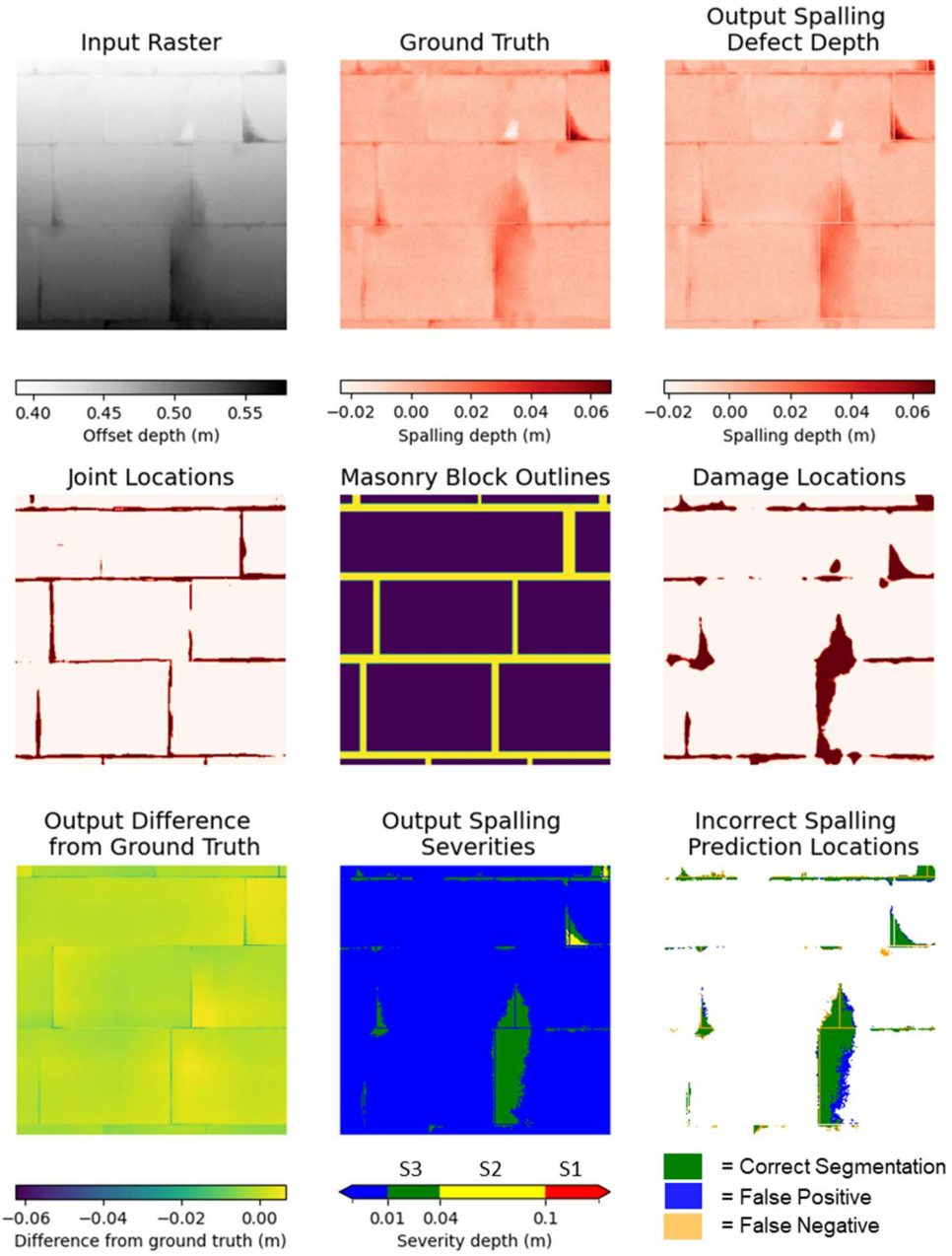


Figure 9.4: Section of T1 with good algorithm performance. This figure shows the output of each stage of the spalling severity segmentation workflow. It also shows the difference between the manually generate ground truth surface damge depth map with that produced by the workflow and highlights the locations of false positive and false negative spalling location predictions relative to the output of the manual analysis.

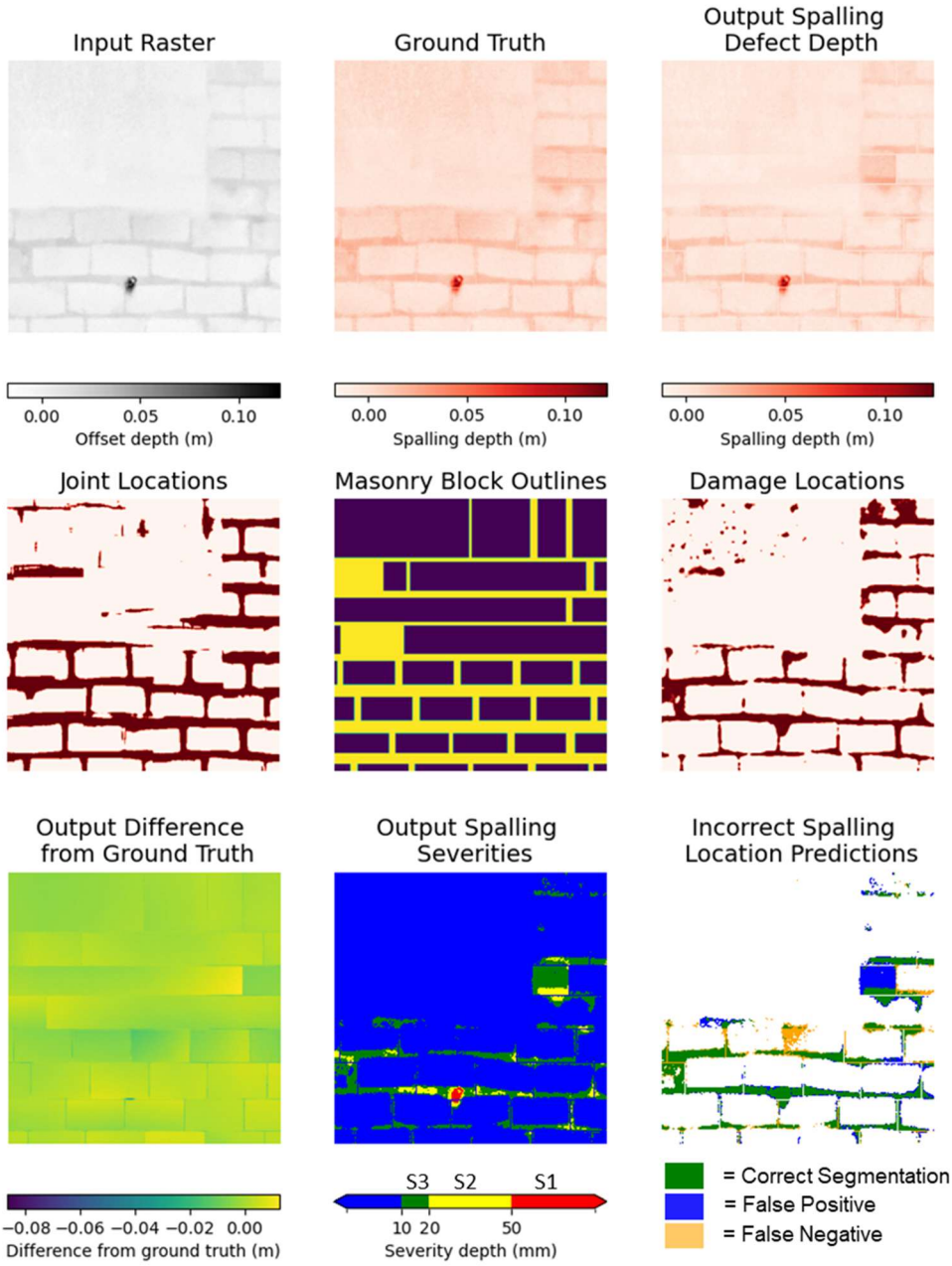


Figure 9.5: Section of T2 showing poor algorithm performance. This figure shows the output of each stage of the spalling severity segmentation workflow. It also shows the difference between the manually generate ground truth surface damge depth map with that produced by the workflow and highlights the locations of false positive and false negative spalling location predictions relative to the output of the manual analysis.

9.2. Real world application to T5

The Brockville tunnel (T5) introduced in Chapter 4 was used as a case study application for the workflow. The automated spalling and deformation detection workflow was applied to a section of the tunnel obtained during a single 32-minute duration high resolution lidar survey. The section's length was limited to only cover areas where the maximum average point spacing was 1mm. The scan was recorded from a chainage of 108m from the south end of the tunnel. The masonry at the scanned section was constructed from brick and noted as being in generally a good condition with some of the mortar having been repointed during a 2016 refurbishment of the tunnel. However, there are multiple areas with deep mortar loss where there has been no recent repointing. The masonry surfaces were quite rough with irregular profiles in places, although only a few areas of deep (>20mm) spalling were noted during the site visit. Two adjacent blocks on the east facing sidewall had extensive deep spalling.

Near the center of the section is an approximately 3m long area of masonry following a slightly different profile at the crown to the rest of the lining. The masonry in this area is not bonded with the masonry in the rest of the section and shows a construction joint at either end and beneath it at the springing line. The reason behind the unusual construction of this feature is unknown. It may have been constructed to infill a collapsed section of the crown or an open shaft. The tunnel also contains 0.2mx0.3m cutouts in the lining, mostly at regular 1.25m intervals along both the east and west facing sidewalls. The cutouts are located immediately below the springing line and were likely used during construction to affix a wooden temporary support trestle for supporting unfinished masonry lining of the arch area. There are also smaller cut outs approximately halfway up the sidewalls that were likely installed for drainage

of the lining. A visualization of the high-resolution scan is shown in Figure 9.6 alongside two photographs of the lining highlighting key features within the analysed section.

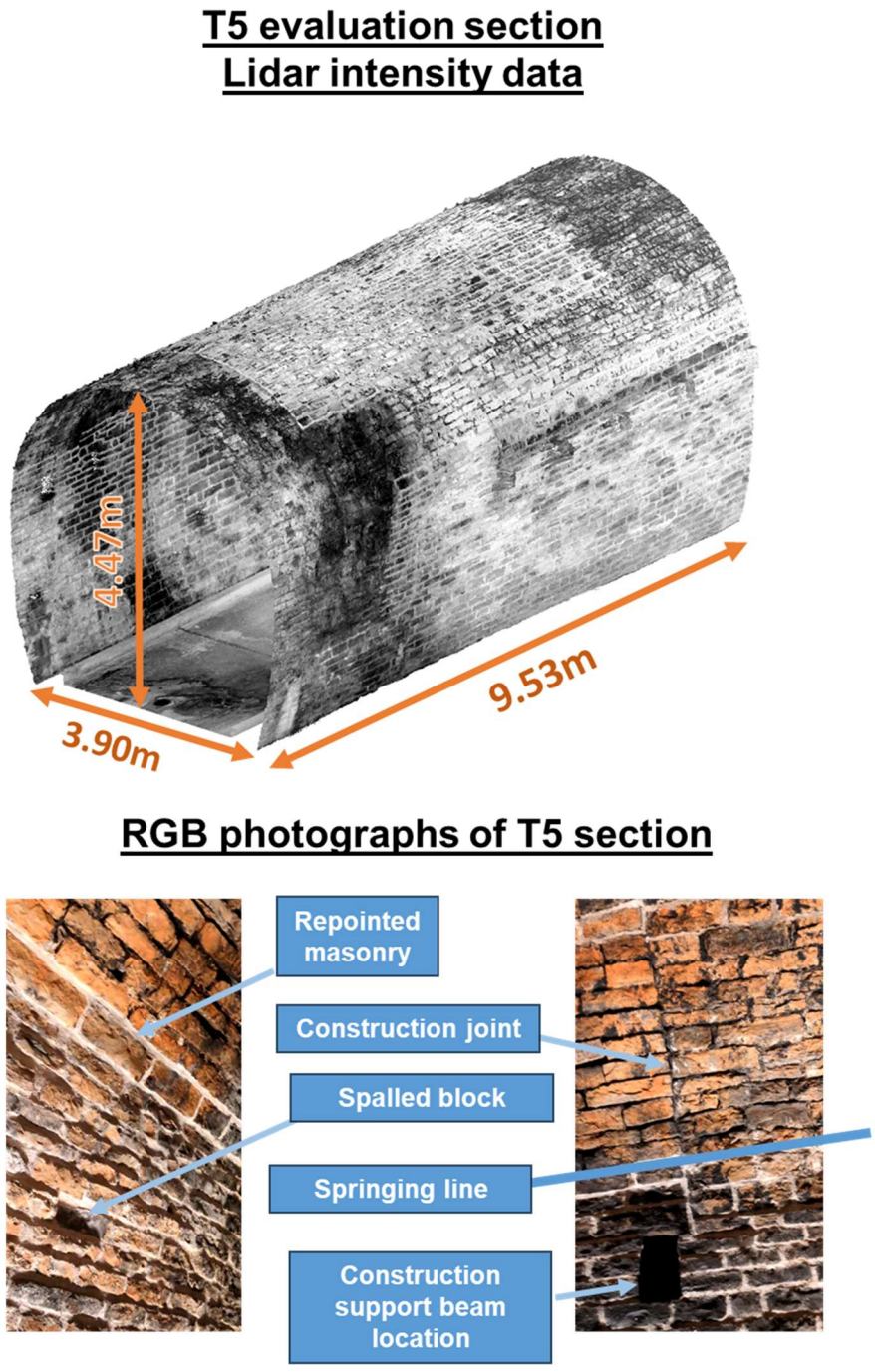


Figure 9.6: Visualisation of the section of T5 selected for evaluation of the workflow. Photographs of the tunnel lining within the section highlight typical features present.

9.2.1. Workflow runtime

The automated spalling and deformation detection workflow was applied to the section of tunnel T5 to evaluate the runtime performance of the workflow. The input high resolution point cloud was spatially subsampled with different minimum point spacings to determine the impact of point cloud density on runtime. The workflow was applied using both the best performing U-Net Mit_b5 model and the smaller Mobilenet_v2 model for masonry joint semantic segmentation. Inference with the Mit_b5 model requires a specialist GPU with at least 10Gb of VRAM which most business workstations will not contain. The lighter but lower performing Mobilenet_v2 encoder will operate with the lower power GPUs typically present on mid-range laptops that are often used by tunnel engineers on site. Figure 9.7 shows the runtime by workflow stage. The performance of each stage was roughly proportional to the density of the input point cloud, as was the overall runtime shown in Figure 9.8. For most models, the rasterising step unexpectedly took the longest to evaluate. Profiling the rasterization function shows that this is due to the pixel binning and interpolation procedures iterating over each pixel in turn within the target image. A substantial speed up could be achieved if the operations were vectorized, although this is challenging due to the unordered nature of the input point cloud. As expected, the Mobilenet_v2 encoder-based network was substantially faster to apply at inference time than the larger Mit_b5 model for each point density. However, as shown in Figure 9.8, joint segmentation using the Mit_b5 encoder did not have a substantially adverse effect on the overall workflow runtime on device L2. It is recommended to apply the Mit_b5 model when sufficient GPU capacity is available given the importance of an accurate joint segmentation output to the workflow’s performance.

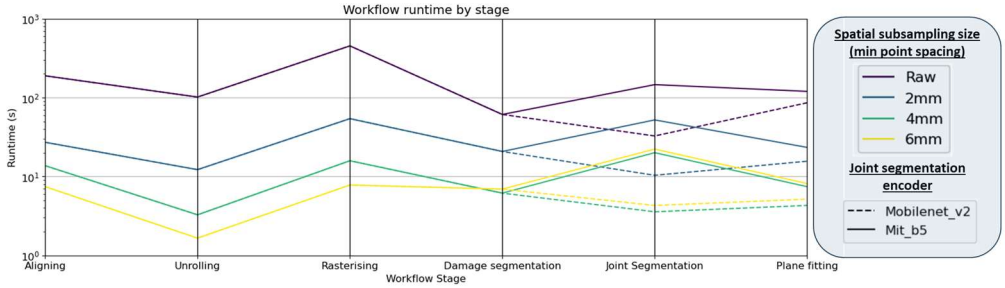


Figure 9.7: Runtime performance by workflow stage and point cloud density. The lower density point clouds were subsampled from the raw point cloud with a greater minimum point spacing.

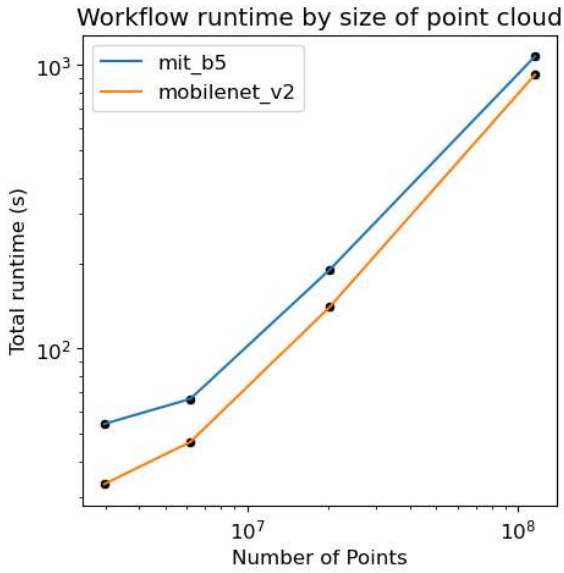


Figure 9.8: Overall workflow runtime by point cloud density. Runtime with the Mit_b5 model for masonry joint segmentation is compared against that when the smaller mobilenet_v2 model is used.

9.2.2. Qualitative performance analysis

The output of the joint segmentation neural network on the rasterized depth map of T5 is shown for different point cloud subsampling spacings in Figure 9.9. The Mit_b5 based network applied to the raw, unsampled point cloud shows few areas of clear false negative segmentations and the segmentation is finer than for the subsampled clouds. Fewer blocks are identified as the

detail of the input point cloud is decreased and there are more large gaps in the generated segmentation map. The Mit_b5 model's output for the point cloud with a 2mm point spacing shows a more regular and less noisy output than that on the raw, unsampled cloud. As the point clouds used for training the neural network were subsampled at a 4mm point spacing, it is possible that the network is confused by small previously unseen features on the masonry surfaces in the raw cloud that are not visible at point spacings greater than 2mm. This may cause false positive joint segmentations within the raw cloud's output. As a result, the Mit_b5 network applied to the 2mm point spacing cloud was chosen for analyzing the workflows spalling output.

Comparison of each of the Mit_b5 outputs with the Mobilenet_v2 equivalents show a broadly similar pattern. Gaps in the Mobilenet_v2 output are in similar locations to gaps in the Mit_b5 output where the masonry surfaces are rougher and the segmentation is more challenging. However, closer inspection shows that the joint connectivity is generally worse in the Mobilenet_v2 output than the Mit_b5 one and individual blocks are often not fully enclosed where they are in the equivalent Mit_b5 output. This is particularly prevalent in the 2mm subsampled output.

Figure 9.10 shows the block surface damage depth map alongside identified radial tunnel deformations and the locations of severe (>20mm depth) spalling. The radial deformation map is visualized reprojected onto the lining point cloud in Figure 9.11 and shows a change in profile of the tunnel of around 100mm between either end of the section. Inspection of the relined section of the arch shows that its radius is similar to the rest of the tunnel section. However, the relined section deviates outwards at its base on the arch springing line and at the circumferential construction joints either side. It is unknown whether the relined section's profile at the springing line has

deviated since construction or was designed as such, although it is possible that it was initially constructed with a slightly larger profile, but has since endured inwards deformation at the crown.

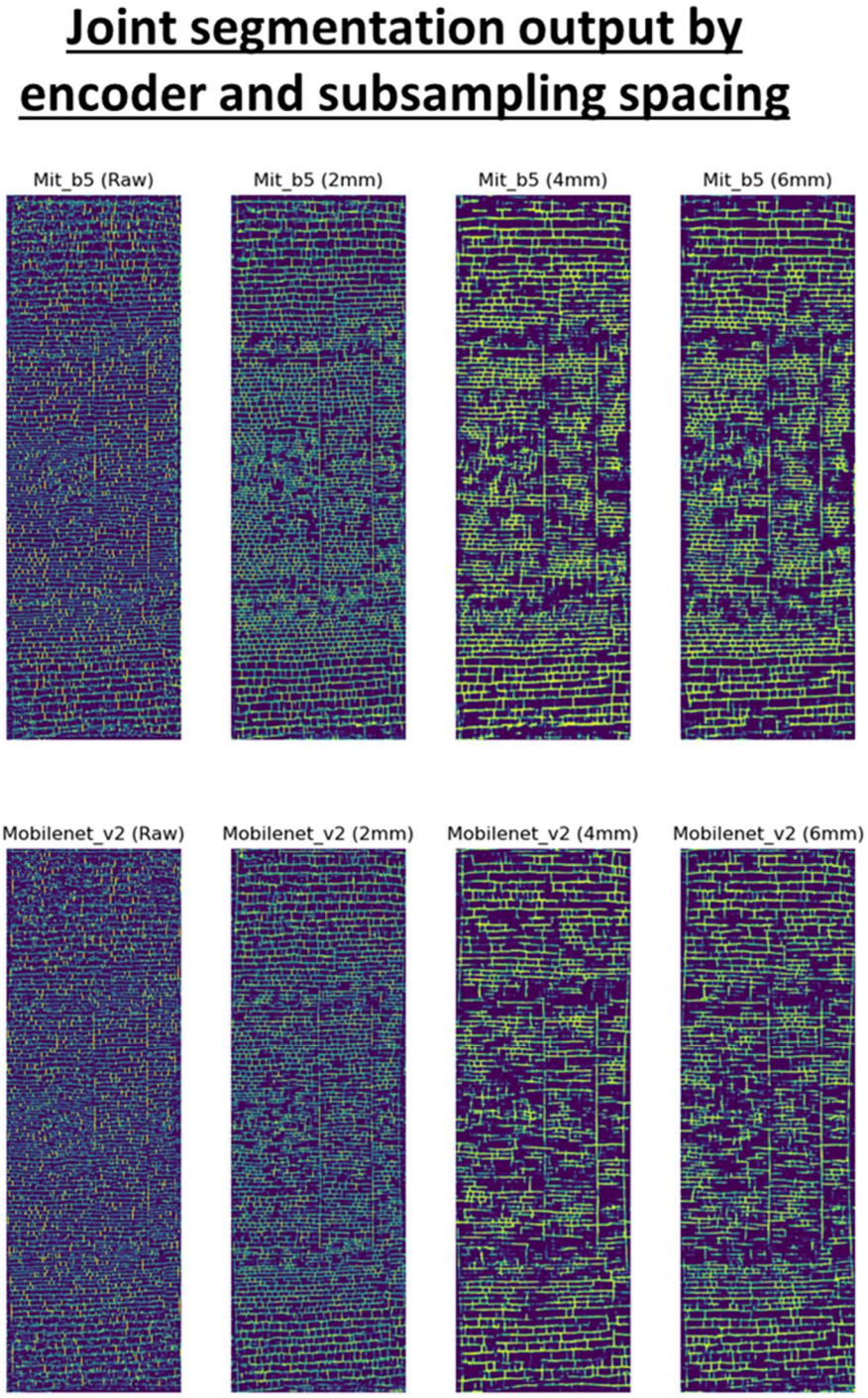


Figure 9.9: Joint segmentation outputs showing the impact of encoder selection and input point cloud density.

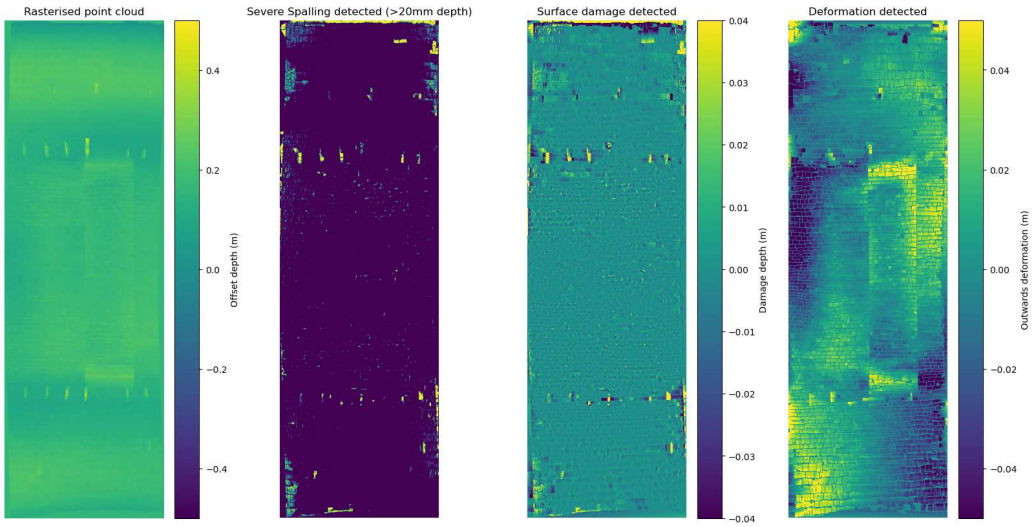


Figure 9.10: Surface damage depth and relative radial lining deformation outputs of the proposed automated workflow. A map of the locations where the more severe severity level S1 and S2 spalling (greater than 20mm depth) have been segmented is also shown.

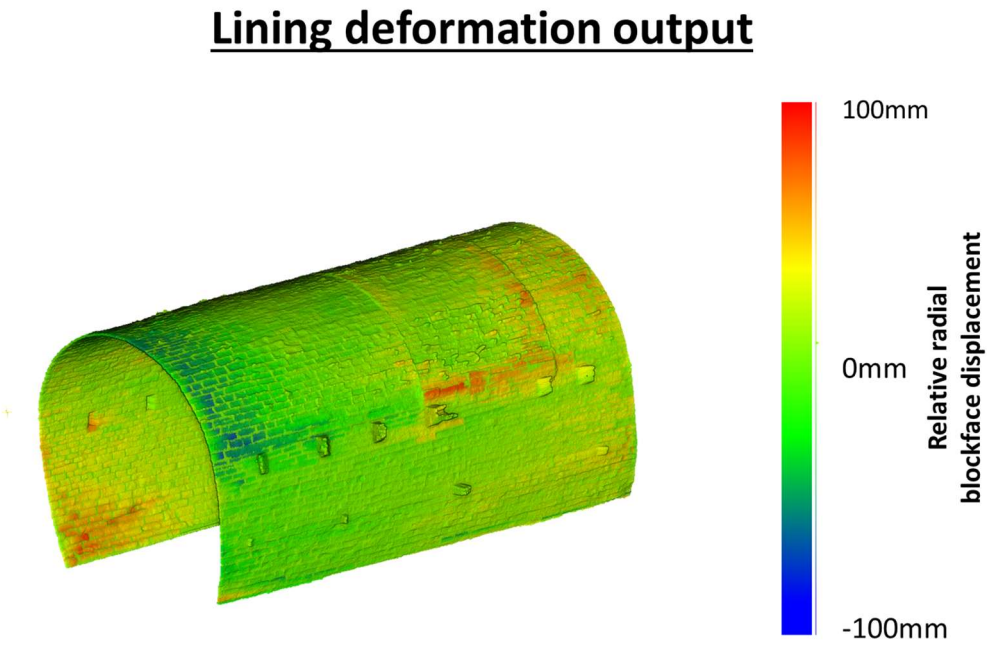


Figure 9.11: Relative radial lining deformation outputs of the proposed automated workflow visualised by reprojection onto the input point cloud.

9.2.3. Surface defect comparison

The automated spalling segmentation output can be analysed through comparison against manually defined spalling maps and by inspecting any changes to the point cloud that have occurred since the initial 2012 lidar survey of the tunnel. The 2025 point cloud was manually aligned with the 2012 scan by matching the locations of 10 points selected on each cloud. These points must be on surfaces that have not been damaged or refurbished between the scans. Surface deposits have also built up on the lining surface between the two scans, so accurate alignment of the clouds was challenging. The offset of each point on the aligned 2025 scan from their closest location on the 2012 scan was then calculated and the value was assigned to the respective point within the 2025 scan as a scalar field. As the incident lidar pulses will not reach exactly the same locations on the tunnel lining in different scans, if the offsets are calculated on a point-to-point basis, then there will always be an error in the calculated point offsets proportional to the point spacing of each scan. To reduce this error, the surface around each point on the 2012 scan was modelled using a best fit plane between the points' 6 nearest neighbor points. The offset of the 2025 scan was calculated from each point to the nearest location on this fitted surface. Nevertheless, small errors in the offset distance will still be present where the linear surface fit poorly characterizes a complex surface geometry and if the cloud is incorrectly aligned.

Using the procedure described in section, a manual surface defect assessment of the 2025 point cloud was also conducted. This involved manually fitting planes representing the undamaged masonry surface and applying offsets and piecewise curvature to match them against the irregular tunnel surface. Multiple fitting algorithms were used to aid the procedure. The

offset of the point cloud from these surfaces was used to create the surface damage depth map. The geometry of the rectangular temporary support openings in the sidewalls were not accurately modelled causing circles of incorrectly identified damage to occur around them.

The Manually identified damage map is compared against the change detection results in Figure 9.12 to identify which damages have occurred within the 13 years between the 2012 and 2025 scans. These recently occurring defects are the most important to identify as they signal increasing deterioration of a lining area, so it is important that they are identified by the automated workflow. The following important features within the change map are highlighted in Figure 9.12:

A) Some large offsets occur at joint areas even where there has been no substantial mortar loss. These occur for two different reasons. In this area, the masonry has been repointed and gaps in the mortar have been filled in during the tunnel's refurbishment. In other areas where there has been no repointing, differences in the exact depth of deep mortar loss may occur due to differing locations of the lidar scanning station. Laser pulses arriving at oblique angles to the mortar loss will only record the edges of the damage and not reach the deepest point.

   While different mortar conditions may impact the performance of the joint fitting neural network, the spalling assessment workflow should focus on damage to the blocks and not show these changes in either the spalling or lining deformation output.

B) In this location two substantially spalled blocks have been replaced during the refurbishment works, so display a good condition in the new survey.

C) & D) These areas show higher severity (S1 and S2 level), deep spalling that has formed since the 2012 survey. Detecting these forms a vital test of the automated workflow's performance.

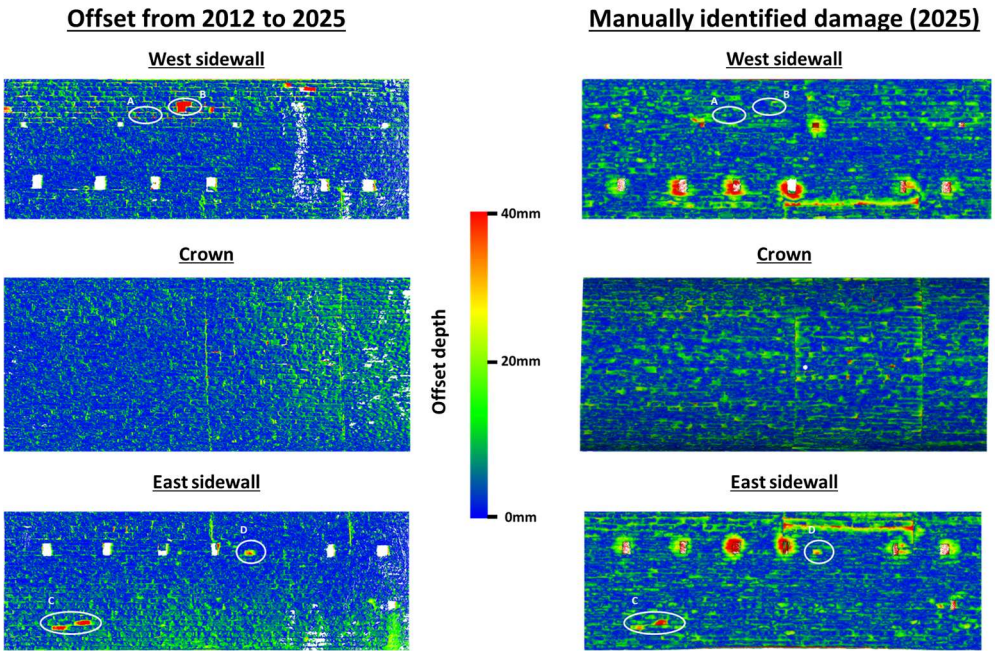


Figure 9.12: Views of the tunnel lining point cloud facing in different directions. A comparison of damage depth outputs between the change detection approach and the manual lining fitting method is shown. Key features are circled and labelled in white.

A comparison was then conducted between the manually generated defect depth map and the automated workflow output. Overall, the manual workflow highlights a substantially larger area of low severity (<20mm depth) spalling than the automated workflow. The masonry within this part of the tunnel has a very rough and uneven surface, so it is subjective whether these deviations were caused by spalling or form the as built masonry surface. The small surface deviations do not have a structural impact on this lining, as the level of mortar loss in some areas forms a larger impact in decreasing the effective lining thickness. Given its context, the lowest severity (<20mm depth) surface

damage would probably be ignored by a human assessor during a routine visual inspection of this section. Representative areas of deeper, higher severity S1 and S2 spalling are highlighted in Figure 9.13 and examined as follows:

A) , B) & C) These areas of deep spalling are correctly identified by the workflow. However, there are large areas of false positive detections adjacent to A and C caused by poor joint segmentation performance in these areas. This is due to very rough block surfaces and irregular block orientations making it difficult to either positively identify or infer joint locations from depressions in the blocks.

D) These apparent false positives occur on the boundary of a construction joint. Closer inspection of the area reveals masonry spalling adjacent to the joint boundary that is not identified in the manual analysis. The exact location of the spalling is incorrectly identified, as it is not picked up by the damage segmentation neural network and so not masked out during plane fitting. As a result, the plane is fitted through the spalled area leading to a false positive in the adjacent part of the block which brings an assessor's attention to the region. This situation still serves to aid an assessor in highlighting a region that requires further manual analysis

E) This spalling is correctly identified

F) The manual damage detection method shows substantial false positive damage adjacent to construction joints such as at this location, in addition to forming circles of hallucinated damage around other lining deviations such as at openings for drainage and for those for holding construction supports. These false positives form from inaccuracies in the manually defined undamaged surface that is

challenging to model at changes in the lining geometry. In general, the automated spalling assessment workflow correctly avoids detecting damage at these locations. At location F, the workflow correctly identifies a small area of spalling forming at the top of masonry underneath a construction joint. This area could only be manually verified through visual inspection of the point cloud as it is not well defined in the manually generated damage map.

G) There are four smaller areas of deeper spalling here occurring at masonry block edges. These are not clear in the automated workflow's output, as the deep spalling adjacent to joints are misidentified during the joint semantic segmentation step as forming part of the masonry joints, so are ignored.

H) Having formed since the 2012 scan of the tunnel, this area of deep spalling was highlighted in both the change detection map and the manually created damage map. Its location was correctly identified by the automated spalling segmentation workflow, although its depth was slightly underestimated. This was caused by the plane fitting procedure fitting part of the block surface to spalled parts of each block due to under segmentation of the damage by the damage segmentation neural network. Nevertheless, this area of spalling is more clearly highlighted in the automated workflow's output, than in the manual output due to the substantial volume of lower severity spalling in the manually generated depth map.

The left side spalled area was correctly segmented by the automated workflow, although again with its depth was underestimated. The automated workflow produced a false positive area of spalling on the right. This is due to false negative segmentations in the output of the joint segmentation neural

network causing adjacent blocks to be identified as one. The radial offset of one block from the other is therefore misidentified as surface damage. The full spalling severity output of the workflow is shown in Figure 9.14.

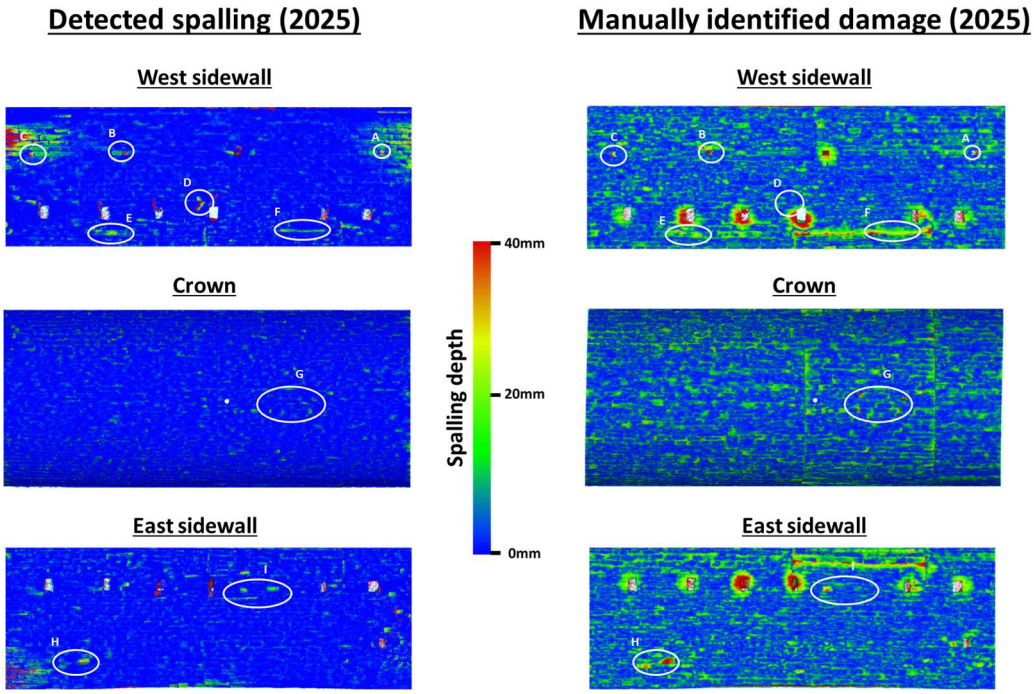
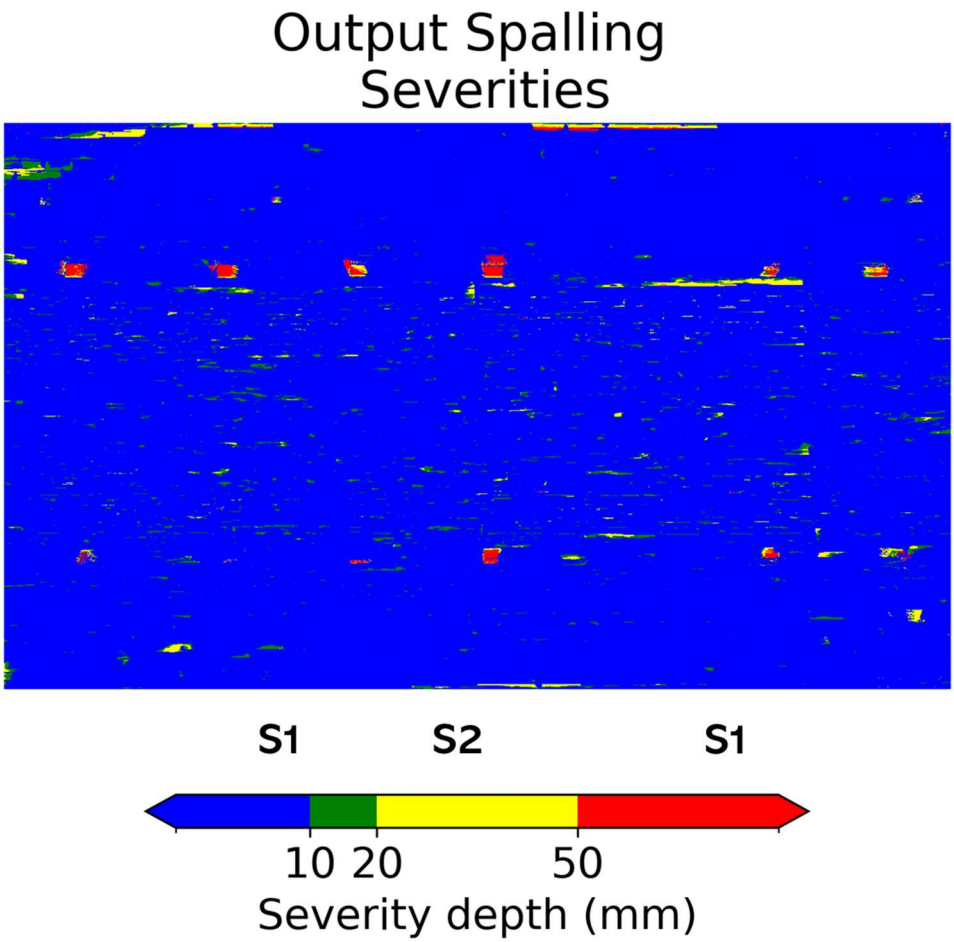


Figure 9.13: Views of the tunnel lining point cloud facing in different directions. A comparison between automated and manually generated spalling depth maps of T5 projected onto the cloud are shown. Key features are circled and labelled in white



Defect code	Depth of Spalling
S1	> 50mm
S2	> 20mm and ≤ 50mm
S3	> 10mm and ≤ 20mm
No Concern	≤ 10mm

Figure 9.14: Spalling severity segmentation output on the selected section of the Brockville Tunnel (T5). The unrolled and rasterised spalling map is shown.

9.3. Discussion

This section evaluated the overall performance of the spalling severity segmentation workflow on a test section of tunnels T1 and T2, with the deep learning parts of the workflow having been trained on separate sections of the same tunnels. A quantitative analysis of the spalling segmentation performance was conducted that demonstrated excellent performance on the section of T1, particularly for the most critical highest severity spalling. Performance was adequate on tunnel T2 although it was limited by the joint segmentation output due to the lower point cloud resolution per block in the smaller brick masonry.

Performance was then evaluated without further training on Tunnel T5. Tunnel T5 was more challenging to analyse due to its rougher masonry surface and a greater number of unusual masonry geometries, such as around the drainage holes and at the construction joints. Small areas of spalling at masonry block edges were sometimes missed, as they were misidentified as part of the masonry joint during the masonry joint segmentation stage. The workflow successfully identified the highest severity (>50mm depth) spalling that had formed within the last 13 years since the 2012 survey of the tunnel, demonstrating its effectiveness with the Other_all training method outlined in Chapter 7.

The runtime of the workflow was also evaluated with different point spacings. For device L2, the joint segmentation stage had a relatively small impact on the overall runtime given the criticality of obtaining good joint segmentation performance in producing an effective spalling severity map. As a result, it is recommended that the computationally heavier Mit_b5 encoder is used in the joint segmentation neural network where possible with the available compute resources.

A key takeaway of this section is the impact of scan resolution on workflow performance. A minimum point to point spacing on the masonry lining of 2mm for brick masonry and 4mm for larger stone masonry is recommended. The ability to obtain higher resolution lidar surveys is generally limited by the time available on site and a circa 5mm point spacing is usually specified for manually conducted condition assessments. However, with newer scanners achieving faster scan rates, it is likely that 2mm point spacing surveys will become routine.

Overall, the workflow shows excellent performance at identifying the most structurally impactful high severity spalling. Where errors occur due to the block face plane being incorrectly fitted, false positive spalling will be identified on a different part of the block. This type of error is easily spotted during manual verification of the workflow's outputs.

Where there is no unusual lining geometry, the workflow is shown to be robust to localised joint segmentation errors leading to multiple blocks being misidentified as one, causing only a small change in the severity of detected spalling. Large offsets of blocks from each other, such as at construction joints, generally lead to better joint segmentation performance. As a result, larger errors in the segmented severity caused by offsets between incorrectly connected neighbouring blocks are unusual.

10. Concluding remarks

This thesis aimed to demonstrate the potential of machine learning based masonry lined tunnel condition assessments by providing a proof of principle workflow for a fully automated method of spalling detection, localisation and severity classification in masonry lined tunnels. The thesis contains multiple novel contributions forming the first study to:

1. Present a method for automatic segmentation of masonry spalling severity on real world masonry linings, demonstrating the benefits of integrating automation through machine learning with a geometric spalling depth calculation that guarantees a physical basis to the segmentation output.
2. Comprehensively compare the performance of different high-level approaches to masonry joint semantic segmentation from 3D point clouds, comparing using 2D computer vision methods with a single surface projection approach against 3D pointwise CNNs and decision-tree based methods.
3. Evaluate the impact of training regime on semantic segmentation performance when using diverse real-world datasets, statistically demonstrating that training with a higher volume of data that has closer features to those in the target test dataset both improves performance and reduces the variation in trained neural network performance.
4. Show how uncertainty quantification methods can be used when automated machine learning based workflows are used in real world condition assessments to help identify the causes of poor segmentation performance.

5. Generate a metric for directly assessing masonry block instance segmentation performance. The proposed BWIoU can be used with semantic segmentation models trained for masonry joint segmentation to evaluate performance on the downstream masonry block instance segmentation task.

The workflow proposed in this thesis uses a 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, the proposed 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 during 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 undamaged 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 evaluated on data from two different tunnels in the UK and applied “off-the-shelf” to the Brockville tunnel in Canada. It was shown to perform particularly well overall on the stone lined tunnel test data. High severity spalling was 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 lower accuracies in the benchmark manual assessment results.

Following application of the automated workflow, each of the most severe areas of spalling will need to be visually inspected either on the point cloud or on site by an engineer to determine their exact impact on the tunnel’s serviceability and to recommend specific repair works. The proposed workflow aims to focus human assessors’ attention on the most critical areas of spalling, saving time against a fully manual visual inspection of the structure. While a few false positive segmentations do occur, these can be quickly discarded on visual inspection.

With regards to the key factors currently limiting condition assessment performance that were highlighted in section 1.3, the proposed workflow brings the following benefits to tunnel condition assessments:

- The workflow creates a comprehensive spalling severity map of the target tunnel from a lidar survey. Manually recording the location and

extent of spalling in the tunnel with the same degree of thoroughness would not be possible during the course of a routine inspection.

- Future developments in mobile lidar technology may enable 3D point clouds to be collected directly from in service trains with the same level of accuracy as that achieved by the TLS surveys used in this study. Combined with the proposed assessment workflow, this would remove the need for as many site inspections, reducing labour costs and improving health and safety outcomes with less time on site and fewer railway closures required.
- Analysis on the Brockville tunnel in Chapter 9 showed that manual calculation of spalling depth from 3D point clouds is time consuming and challenging. It is not usually feasible to investigate every part of a tunnel lining 3D point cloud in depth, leading to rough modelling of the deformed intrados surface. Manually created surface damage depth maps therefore often have many errors, so in some locations the automated method performs better than the manual digital one. In addition, the manual digital method does not distinguish masonry block spalling from mortar loss, unlike the automated workflow which identifies mortar and masonry joint locations.
- The IoU performance of the spalling severities segmented on the test data section of tunnel T1 in Chapter 9 were greater than 0.5 for each severity level. This demonstrates that on tunnels where the joint and damage segmentation is adequately accurate, high spalling severity segmentation performance can be achieved throughout.
- Analysis of the workflow on test data from tunnels T1 and T2 in Chapter 9 showed that the total area of spalling segmented was only 31.4% and 8.0% less than that within the test data. Given that the difference between the area of spalling that would cause different

types of remediation work to be recommended vary by orders of magnitude, the amount of spalling segmented by the automated workflow successfully gives an accurate enough estimate to gauge the health of a structure.

- The workflow is able to run effectively on standard office computer hardware (when the Mobilenet_v2 encoder is used for joint segmentation), so there is no increased cost to applying the workflow if 3D tunnel point cloud data is already being collected.
- Individual masonry block instances are segmented as part of the proposed automated workflow. This provides an accurate documentation of the tunnel lining geometry which could be useful within further analysis such as numerical modelling of the lining stability or for use in tunnel BIM.
- The multi-step nature of the proposed workflow makes it more explainable than a single step machine learning-based solution. It is easy to investigate what are the causes of incorrect segmentations when they occur. The performance of the workflow is also largely reliant on the performance of the joint segmentation neural network, the output of which is straight forward to inspect for block segmentation errors.
- The use of geometric plane fitting and a physical geometric distance to calculate the spalling depth map and threshold spalling severities makes the spalling depth calculation more robust to small abnormalities on the tunnel lining that may produce isolated areas with low masonry damage and joint segmentation performance. Furthermore, there is no subjectivity in the depth calculation, unlike during manual visual inspections where it is not feasible to conduct exact spalling measurements.

- Once the CNN parts of the workflow are trained, the proposed workflow will produce the same result every time it is applied to the same data. The results are therefore reproducible.
- The workflow produces a 3D point cloud of the target tunnel lining with the spalling locations labelled. This enables easy communication of exact damage locations to maintenance teams. 2D unrolled raster images are also produced which can be used directly in maintenance reports.
- A secondary output of the workflow produces a relative radial displacement depth map of the tunnel lining. This helps communicate the location of a tunnel's lining intrados surface to a tunnel assessor so that they can more easily identify lining deformations.

In addition to the benefits of the proposed workflow, this thesis investigated two aspects which impact the trustworthiness and reliability of any CNN-based method applied for tunnel lining analysis. Given the range of masonry lined tunnel geometries, material types and damage levels, CNN-based methods achieve different levels of accuracy when applied to different tunnels. It is therefore vital that the outputs of a CNN-based method are reliable and can be trusted when applied in a new tunnel situation.

An investigation was conducted into the variability in the joint segmentation neural network's outputs when trained using different regimes. In general training with more data improved performance and produced more consistent results, although additional tunnel specific training was shown to lead to more substantial performance improvements per volume of training data. Sufficient performance was achieved when a model was trained only on a small section of the target tunnel after being pretrained on ImageNet, highlighting the effective transfer learning performance of the U-Net

architecture. This demonstrates that it may still be worth an organisation using the proposed workflow for tunnel condition assessment even when they do not have a large database of previously assessed tunnels to use as training data. It was also shown that topological loss functions, such as CLDice can slightly improve performance of CNNs trained for masonry joint segmentation. However, CLDice was also more challenging to train. When applied instead of a softdice loss function, CLDice occasionally failed to converge during training.

Overall, the performance of a CNN-based damage segmentation method is shown to be strongly dependant on quantity and similarity of training data to the target test dataset. As a result, it is necessary in practice for an engineer to visually assess the similarity of the target tunnel's lining to the training data to evaluate whether the trained model will have previously seen equivalent lining features in its training data and be able to generalise well to the target tunnel. Based on the analysis conducted in Chapter 7, it is expected that the proposed models would perform adequately on most stone and brick tunnel linings when trained on tunnels T1, T2, T3 and T4 then applied to unseen tunnels with low levels of damage and typical masonry designs. Performance of the Single and Full_all training regimes compared to the Other_all regime show that adding targeted sections of lining with similar features to the test data into the training data can enable adequate performance to be achieved when a lining situation is not present in the initial training data.

Uncertainty quantification methods were investigated to determine if they could be used to predict workflow performance on unseen tunnels. The study demonstrated that Test-Time Augmentation and Monte Carlo Dropout can be used to indicate Aleatoric and Epistemic uncertainty respectively in a CNN's segmentation. A small correlation was found in some locations

between the Area Variation Coefficient used to quantify the segmentation's uncertainties and the local joint segmentation performance of a trained CNN. However, using both test-time augmentation and Monte Carlo Dropout for uncertainty quantification was shown to be more useful as part of a toolkit for users of the proposed automated spalling segmentation workflow. Area Variation score maps can be used to help an assessor interrogate the output of the masonry joint segmentation CNN in known areas that it performs badly to better understand the cause of poor performance. This can help a practitioner to make a decision over whether more general or tunnel specific training of the CNN is required to improve segmentation performance.

10.1. Key achievements

The outputs of the studies presented in this thesis meet the project objectives outlined in section 1.2 as follows:

1. An automated method was created for generating masonry tunnel lining spalling severity maps in line with Network Rail standards. It was shown to be faster and less labour intensive to apply than a typical manual analysis of a tunnel lining 3D point cloud.
2. The performance of the proposed workflow was quantified. It showed excellent performance on stone lined tunnels and adequate performance on brick lined ones. A qualitative analysis showed that the automated workflow was able to positively identify all areas with the highest severity spalling and provide a good estimate of the total area of low severity spalling in a tunnel.
3. The automated workflow could operate directly on routinely collected 3D point cloud data obtained by lidar. Analysis of the workflow on the Brockville tunnel showed how the workflows' outputs can be easily

verified by a human assessor to ensure that spalling was correctly identified.

4. The impact of training data type and volume on generalization performance was analysed. The study showed that even when trained with the same data and hyperparameters, performance of the joint segmentation stage will be variable and dependent on the volume and variety of training data. However, training with more data was shown to produce models with less variability in their performance.
5. While there is still further work that would need to be conducted to fully understand the proposed workflow's limitations, Test-Time Augmentation was shown to give an indication of the aleatoric uncertainty in a CNN's prediction. The Area Variation Coefficient can be used to identify lining areas where a trained masonry joint segmentation CNN will have more difficulty due to lower clarity in the input data.

10.2. Limitations

Despite these achievements, the following key limitations have been identified with the proposed workflow:

- The workflow is limited by the performance of the neural network used for masonry joint segmentation, which in turn is limited by training data availability. If joint closure around each block is not achieved by the joint segmentation CNN, then spalling segmentation performance substantially decreases.
- The best performance was achieved using a lidar point spacing of 2mm on the tunnel lining. Using standard TLS equipment, it is time

consuming and non-standard to collect data this densely. Until laser scanning speed increases, additional costly data collection shifts may be required to collect enough data of a target tunnel.

- The output of the workflow still requires expert interpretation and analysis of the produced 3D point cloud to identify and discard false negative segmentation outputs.
- The best performance was achieved when at least 10m of tunnel specific data was used in addition to a database of other tunnels for training the joint segmentation CNN part of the workflow. However, creating the ground truth data is a time consuming process at around 4 hours for 10m of stone lining. Nevertheless, it is generally still quicker and less subjective to manually create masonry joint map training data for a short section of the tunnel, than to conduct a full manual spalling analysis of a long tunnel (around 3 hours per 20m section).
- It is known that performance decreases when applied to tunnels out of distribution of the training data. However, uncertainty quantification methods provide only an indication of challenging locations and the uncertainty values produced do not directly correlate with workflow performance.

Furthermore, there are also limitations to the extent of the analysis conducted that add uncertainty to the effectiveness of the proposed workflow:

- Manual identification of masonry spalling is a subjective process, but was only conducted by a single human assessor for the creation of ground truth data used in this study. As a result, there may be errors in the ground truth data that cause the recorded performance of the proposed automated workflow to be misleading and biased to a

particular interpretation of the lining’s condition. Ideally, the ground truth data should be formed by a combination of human experts.

- Four tunnels were used for tuning, training and qualitatively testing each part of the proposed workflow. While these showcase a variety of lining types and damage levels, the generalization performance of each trained neural network and the entire workflow cannot conclusively be determined without higher volumes of data.

10.3. Recommendations

The studies within this thesis have aimed to show that when applied appropriately, the proposed workflow is likely to achieve adequate spalling severity segmentation performance on a target tunnel, and equal the output typically achieved by a human assessor. However, before a similar workflow can be applied within industry there are four further areas where 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 four 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. Performance of the workflow need to be analysed in a wider variety of masonry tunnel situations and further development of the workflow is required to enable it to operate on more unusual lining geometries.

Further research needs to be conducted into determining the boundaries of effectiveness of the CNN aspects of the workflow. It would be useful to create

a metric that can be used to assess the likely performance of the automated workflow on a target tunnel without needing to manually produce substantial amounts of ground truth verification data. This would increase the trustworthiness of the method.

Finally, research needs to be conducted into determining and fulfilling practical considerations to further reduce the required operator input. The workflow needs to be assessed by a third party during a real-world tunnel condition assessment to investigate its usability.

10.4. Publications

Parts of the research presented in this thesis have been published in

Journals:

Smith, J., Paraskevopoulou, C., Cohn, A.G., Kromer, R., Bedi, A., Invernici, M., 2024. Automated masonry spalling severity segmentation in historic railway tunnels using deep learning and a block face plane fitting approach. *Tunnelling and Underground Space Technology 153, 106043*. https://doi.org/10.1016/j.tust.2024.106043

Smith, J., Paraskevopoulou, C.,. 2025. Practical assessment of masonry tunnel joint segmentation using topological machine learning. *Civil Engineering Design 7, 93–110.* https://doi.org/10.1002/CEND.202400049

Smith, J., Paraskevopoulou, C.,. 2025. Uncertainty Quantification to Assess the Generalisability of Automated Masonry Joint Segmentation Methods. *Infrastructures (Basel) 10, 98.* https://doi.org/10.3390/INFRASTRUCTURES10040098

Journals in preparation:

Smith, J., Paraskevopoulou, C., Diederichs, M.,. Real world performance of AI-driven tunnel condition assessment on a 19th century tunnel. To be submitted to: *Automation in Construction*. [Paper in preparation]

Conference proceedings:

Smith, J., Paraskevopoulou, C., Bedi, A., Invernici, M.,. 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.

Smith, J., Paraskevopoulou, C.,. 3D Deep Learning for Segmentation of Masonry Tunnel Joints. *Structural Integrity Procedia, proceedings of the 7th International Conference on Smart Monitoring, Assessment and Rehabilitation of Civil Structures*, SMAR 2024, Volume 64, Pages 220-227.

Conference publications in preparation:

Smith, J., Paraskevopoulou, C., Diederichs, M., Automated damage assessment of masonry lined tunnels using machine learning on lidar data. Submitted to: WTC 2026 of Word Tunnelling Congress on .Connecting Communities Through Underground Infrastructure, 16-21 May 2026, Montreal, Canada. [Abstract accepted, Paper in preparation].

Paraskevopoulou, C., **Smith, J.,** Diederichs, M., Long-Term Deterioration and Structural Analysis of Masonry-Lined Tunnels: A Case Study of the Brockville Tunnel. Submitted to: International symposium preservation of monuments & historic sites. TC301 International Symposium, 16-18 September, Athens, Greece. [Abstract accepted, Paper in preparation].

Paraskevopoulou, C., **Smith, J.,** Diederichs, M., Tunnels Through Time: The Rehabilitation and Repurposing of Victorian Era Rail Infrastructure.

Submitted to: WTC 2026 of Word Tunnelling Congress on .Connecting Communities Through Underground Infrastructure, 16-21 May 2026, Montreal, Canada. [Abstract accepted, Paper in preparation].

The research has also been presented in

External presentations:

Smith, J., Paraskevopoulou, C.,. Synthetically generated spalling can improve the performance of automated tunnel lining masonry joint detection. *ISMLG 2023 (4th International symposium on Machine Learning & Big Data in Geoscience)*, 29 August–01 September 2023, Cork, Ireland.

Smith, J., Paraskevopoulou, C.,. A review of the applications of new technologies to condition assessment of tunnels. Presentation in: *British Tunnelling Society Young Members Conference, BTSYM 2023*, September 2023, London, UK

Smith, J., Paraskevopoulou, C.,. Defect detection on masonry lined tunnels using deep learning. Presentation in: *ITA-CET Professors Meeting and Doctoral Students Conference, ITACET 2024*, June 2024, Zurich, Switzerland

Smith, J., Paraskevopoulou, C.,. Automated deformation assessment of masonry lined tunnels. *Special Session on AI-enriched SHM of civil engineering structures, 11th European Workshop on Structural Health Monitoring*, *EWSHM 2024,* 10-13 June 2024, Potsdam, Germany.

Smith, J., Paraskevopoulou, C.,. Assessing the potential for automated condition assessment of masonry lined tunnels. Presentation in: *British Tunnelling Society Young Members Conference, BTSYM 2024*, November 2024, London, UK

References

3DSourced, 2024. 10 Best Photogrammetry Software 2024 [WWW Document]. 3D sourced. URL https://www.3dsourced.com/3d-software/best-photogrammetry-software/ (accessed 8.2.25).

Abraham, N., Khan, N.M., 2019. A novel focal tversky loss function with improved attention u-net for lesion segmentation. Proceedings - International Symposium on Biomedical Imaging 2019-April, 683–687. https://doi.org/10.1109/ISBI.2019.8759329

Abu-Haifa, M., Lee, S.J., 2022. Image-Based Modeling-to-Simulation of Masonry Walls. Journal of Architectural Engineering 28, 06022001. https://doi.org/10.1061/(ASCE)AE.1943-5568.0000569/ASSET/E69F3036-2EC7-48D8-BE99-A6282EBA4684/ASSETS/IMAGES/LARGE/FIGURE14.JPG

Adiga V., S., Dolz, J., Lombaert, H., 2024. Anatomically-aware uncertainty for semi-supervised image segmentation. Med. Image Anal. 91, 103011. https://doi.org/10.1016/J.MEDIA.2023.103011

Agnisarman, S., Lopes, S., Chalil Madathil, K., Piratla, K., Gramopadhye, A., 2019a. A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection. Autom. Constr. 97, 52–76. https://doi.org/10.1016/J.AUTCON.2018.10.019

Agnisarman, S., Lopes, S., Chalil Madathil, K., Piratla, K., Gramopadhye, A., 2019b. A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection. Autom. Constr. 97, 52–76. https://doi.org/10.1016/J.AUTCON.2018.10.019

Ahmed, H., La, H.M., Gucunski, N., 2020. Review of Non-Destructive Civil Infrastructure Evaluation for Bridges: State-of-the-Art Robotic Platforms, Sensors and Algorithms. Sensors (Basel). 20, 1–38. https://doi.org/10.3390/S20143954

Ali, L., Khan, W., Chaiyasarn, K., 2019. DAMAGE DETECTION AND LOCALIZATION IN MASONRY STRUCTURE USING FASTER REGION CONVOLUTIONAL NETWORKS. International Journal of GEOMATE 17, 98–105. https://doi.org/10.21660/2019.59.8272

AnalytixLabs, 2024. Convolutional Neural Networks – Definition, Architecture, Types, Applications, and more [WWW Document]. https://www.analytixlabs.co.in/blog/convolutional-neural-network/.

Antón, D., Carretero-Ayuso, M.J., Moyano-Campos, J., Nieto-Julián, J.E., 2022. Laser Scanning Intensity Fingerprint: 3D Visualisation and Analysis of Building Surface Deficiencies, in: New Technologies in Building and Construction. Springer Science and Business Media Deutschland GmbH, pp. 207–223. https://doi.org/https://doi.org/10.1007/978-981-19-1894-0_12

Ao, X., Wu, H., Xu, Z., Gao, Z., 2018. Damage Extraction of Metro Tunnel Surface from Roughness Map Generated by Point Cloud, in: International Conference on Geoinformatics. IEEE Computer Society. https://doi.org/10.1109/GEOINFORMATICS.2018.8557062

Arega, T.W., Bricq, S., Meriaudeau, F., 2021. Leveraging Uncertainty Estimates to Improve Segmentation Performance in Cardiac MR. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 12959 LNCS, 24–33. https://doi.org/10.1007/978-3-030-87735-4_3/TABLES/2

Asadi Shamsabadi, E., Xu, C., Dias-da-Costa, D., 2022. Robust crack detection in masonry structures with Transformers. Measurement 200, 111590. https://doi.org/10.1016/J.MEASUREMENT.2022.111590

Asgari Taghanaki, S., Abhishek, K., Cohen, J.P., Cohen-Adad, J., Hamarneh, G., 2021. Deep semantic segmentation of natural and medical images: a review. Artif. Intell. Rev. 54, 137–178. https://doi.org/10.1007/S10462-020-09854-1/FIGURES/16

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

Attard, L., Debono, C.J., Valentino, G., Di Castro, M., 2018. Tunnel inspection using photogrammetric techniques and image processing: A review. ISPRS Journal of Photogrammetry and Remote Sensing 144, 180–188. https://doi.org/10.1016/J.ISPRSJPRS.2018.07.010

Ayhan, M.S., Berens, P., n.d. Test-time Data Augmentation for Estimation of Heteroscedastic Aleatoric Uncertainty in Deep Neural Networks.

Babic, L., Pribicevic, B., Dapo, A., 2012. Mobile Laser Scanning (MLS) in transport infrastructure documentation and research. Ekscentar 15, 96–99.

Beaver, P., 1972. A History of Tunnels.

Behbahani, S.S., Golrokh, A.J., Hafiz, A., Wiggenhauser, H., Tabrizi, K., 2024. Utilizing infrared thermography for the condition assessment of tunnel Lining with tiled surface in various temperature conditions. Tunnelling and Underground Space Technology 154, 106093. https://doi.org/10.1016/J.TUST.2024.106093

Bendris, B., Becerra, J.C., 2022. Design and Experimental Evaluation of an Aerial Solution for Visual Inspection of Tunnel-like Infrastructures. Remote Sensing 2022, Vol. 14, Page 195 14, 195. https://doi.org/10.3390/RS14010195

BenTaieb, A., Hamarneh, G., 2016. Topology aware fully convolutional networks for histology gland segmentation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9901 LNCS, 460–468. https://doi.org/10.1007/978-3-319-46723-8_53/FIGURES/3

Bergstrom, A.C., Messinger, D.W., 2022. Understanding the relationship between image quality and convolutional neural network performance. https://doi.org/10.1117/12.2618590 12101, 10–24. https://doi.org/10.1117/12.2618590

Bertels, J., Eelbode, T., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., Blaschko, M.B., 2019. Optimizing the Dice Score and Jaccard Index for Medical Image Segmentation: Theory & Practice. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11765 LNCS, 92–100. https://doi.org/10.1007/978-3-030-32245-8_11

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. Journal of

Computing in Civil Engineering 37. https://doi.org/10.1061/JCCEE5.CPENG-5009

Brackenbury, D., Brilakis, I., Dejong, M., 2019. Automated defect detection for masonry arch bridges. International Conference on Smart Infrastructure and Construction 2019, ICSIC 2019: Driving Data-Informed Decision-Making 3–10. https://doi.org/10.1680/ICSIC.64669.003

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

Bradski, G., 2000. opencv_library. Dr. Dobb's Journal of Software Tools.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32. https://doi.org/10.1023/A:1010933404324/METRICS

Brown, G., Montaigu, T., 2024. Laspy.

Brozovsky, J., Labonnote, N., Vigren, O., 2024. Digital technologies in architecture, engineering, and construction. Autom. Constr. 158, 105212. https://doi.org/10.1016/J.AUTCON.2023.105212

Budd, S., Robinson, E.C., Kainz, B., 2021. A survey on active learning and human-in-the-loop deep learning for medical image analysis. Med. Image Anal. 71, 102062. https://doi.org/10.1016/j.media.2021.102062

Buslaev, A., Parinov, A., Khvedchenya, E., Iglovikov, V.I., Kalinin, A.A., 2018. Albumentations: fast and flexible image augmentations. Information (Switzerland) 11. https://doi.org/10.3390/info11020125

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J., 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics 32, 1309–1332. https://doi.org/10.1109/TRO.2016.2624754

Canny, J., 1986. A Computational Approach to Edge Detection. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8, 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

Cao, R., Ma, M., Liang, R., Niu, C., 2019. Detecting the void behind the tunnel lining by impact-echo methods with different signal analysis approaches. Applied Sciences (Switzerland) 9. https://doi.org/10.3390/app9163280

Carer, M., Diederichs, M., 2012. T021056-G1: Condition Inspection, Brockville Railway Tunnel. Brockville.

Carlson, J.E., Jansson, A., 2021. Modeling of the Stress-Strain Relationship of Rock Bolts from Ultrasound Data. IEEE International Ultrasonics Symposium, IUS. https://doi.org/10.1109/IUS52206.2021.9593859

Casson, M., 2009. The World's First Railway System: Enterprise, Competition, and Regulation on the Railway Network in Victorian Britain. The World's First Railway System: Enterprise, Competition, and Regulation on the Railway Network in Victorian Britain 1–544. https://doi.org/10.1093/ACPROF:OSO/9780199213979.001.0001

Chai, J., Zeng, H., Li, A., Ngai, E.W.T., 2021. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Machine Learning with Applications 6, 100134. https://doi.org/10.1016/J.MLWA.2021.100134

Chaiyasarn, K., 2014. Damage detection and monitoring for tunnel inspection based on computer vision 250.

Chaiyasarn, K., Khan, W., Ali, L., Sharma, M., Brackenbury, D., Dejong, M., 2018. Crack Detection in Masonry Structures using Convolutional Neural Networks and Support Vector Machines, in: Proceedings of the 35th ISARC, Berlin, Germany, ISBN 978-3-00-060855-1, ISSN 2413-5844. pp. 118–125. https://doi.org/10.22260/ISARC2018/0016

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

Chen, J., Jiang, X., Yan, Y., Lang, Q., Wang, H., Ai, Q., 2022. Dynamic Warning Method for Structural Health Monitoring Data Based on ARIMA: Case Study of Hong Kong–Zhuhai–Macao Bridge Immersed Tunnel. Sensors (Basel). 22, 6185. https://doi.org/10.3390/S22166185

Chen, T., Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 13-17-August-2016, 785–794. https://doi.org/10.1145/2939672.2939785

Chen, W., Tan, X.-Y., Yang, J., 2025. Review of state-of-the-art in structural health monitoring of tunnel engineering. Smart Underground Engineering. https://doi.org/10.1016/j.sue.2025.05.004

Cheng, L., Chen, S., Liu, X., Xu, H., Wu, Y., Li, M., Chen, Y., 2018. Registration of Laser Scanning Point Clouds: A Review. Sensors 2018, Vol. 18, Page 1641 18, 1641. https://doi.org/10.3390/S18051641

Chi, Y.T., Chien, C.F., Lin, T. Te, 2003. LEAF SHAPE MODELING AND ANALYSIS USING GEOMETRIC DESCRIPTORS DERIVED FROM

BEZIER CURVES. Transactions of the ASAE 46, 175-. https://doi.org/10.13031/2013.12540

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

Coleman, T., 1981. The Railway Navvies: A History of the Men who Made the Railways.

Copperthwaite, W.C., 1906. Tunnel Shields, and the Use of Compressed Air in Subaqueous Works. Nature 1906 74:1919 74, 348–349. https://doi.org/10.1038/074348a0

Coucher, I., Austin, C., Leunig, T., Smith, R., Currey, A., Sindall, D., Claire, A., Steer, J., Shilton, D., 2008. The Billion Passenger Railway Lessons From The Past: Prospects For The Future.

Cross, M., 2018. Case Study: Benchmarking tunnelling costs and production rates in the UK .

Dais, D., Bal, İ.E., Smyrou, E., Sarhosis, V., 2021. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. Autom. Constr. 125, 103606. https://doi.org/10.1016/J.AUTCON.2021.103606

Davis, A.G., Lim, M.K., Petersen, C.G., 2005. Rapid and economical evaluation of concrete tunnel linings with impulse response and impulse radar non-destructive methods, in: NDT and E International. pp. 181–186. https://doi.org/10.1016/j.ndteint.2004.03.011

Dawood, T., Zhu, Z., Zayed, T., 2017. Machine vision-based model for spalling detection and quantification in subway networks. Autom. Constr. 81, 149–160. https://doi.org/10.1016/J.AUTCON.2017.06.008

de Ville, B., 2013. Decision trees. Wiley Interdiscip. Rev. Comput. Stat. 5, 448–455. https://doi.org/10.1002/WICS.1278

Deep Robotics Ltd., 2025. Deep Robotics' robot dog pioneers smart inspection in Singapore power cable tunnel [WWW Document]. URL https://roboticsandautomationnews.com/2025/01/17/deep-robotics-robot-dog-pioneers-smart-inspection-in-singapore-power-cable-tunnel/88660/ (accessed 6.26.25).

Degrande, G., Clouteau, D., Othman, R., Arnst, M., Chebli, H., Klein, R., Chatterjee, P., Janssens, B., 2006. A numerical model for ground-borne vibrations from underground railway traffic based on a periodic finite element-boundary element formulation. J. Sound Vib. 293, 645–666. https://doi.org/10.1016/j.jsv.2005.12.023

Dekker, B., Ton, B., Meijer, J., Bouali, N., Linssen, J., Ahmed, F., 2023. Point Cloud Analysis of Railway Infrastructure: A Systematic Literature Review. IEEE Access 11, 134355–134373. https://doi.org/10.1109/ACCESS.2023.3337049

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Institute of Electrical and Electronics Engineers (IEEE), pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

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

Do, H.P., Guo, Y., Yoon, A.J., Nayak, K.S., 2020. Accuracy, uncertainty, and adaptability of automatic myocardial ASL segmentation using deep CNN. Magn. Reson. Med. 83, 1863–1874. https://doi.org/10.1002/MRM.28043

Dong, C., Mao, Q., Kou, D., Dai, Y., Xiong, Y., Zhu, X., Li, X., 2022. Dynamic precision surveying of railway tunnels based on MLS with robust constraints of track control network. Meas. Sci. Technol. 33, 085015. https://doi.org/10.1088/1361-6501/AC64E7

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. https://doi.org/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

Dos Santos, K.R.M., Chassignet, A.G.J., Pantoja-Rosero, B.G., Rezaie, · Amir, Onaïa, ·, Savary, J., Katrin Beyer, ·, 2024. Uncertainty quantification for a deep learning models for image-based crack segmentation. Journal of Civil Structural Health Monitoring 2024 1–39. https://doi.org/10.1007/S13349-024-00879-6

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S.,

Uszkoreit, J., Houlsby, N., 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021 - 9th International Conference on Learning Representations.

Drinker, H.S., 1882. Tunneling, Explosive Compounds & Rock Drills, Comprising a Review of Tunneling from the Reign of Rameses II to the Present Time, https://books.google.de/books?hl=en&lr=&id=8SdYAAAAYAAJ&oi=fnd&pg=PA1&dq=history+of+tunnelling&ots=c9YahIARcG&sig=CB4VIICoKP_QQvLLRQRWMFrfVcM&redir_esc=y#v=onepage&q=history%20of%20tunnelling&f=false.

Edelsbrunner, H., Morozov, D., 2014. Persistent Homology: Theory and Practice. European Congress of Mathematics Kraków, 2 – 7 July, 2012 31–50. https://doi.org/10.4171/120

Edmondson, J., 2024. Breaking Down Barriers to AI Adoption: Overcoming Challenges in Implementing Artificial Intelligence - Businesstechweekly.com [WWW Document]. Businesstechweekly. URL https://www.businesstechweekly.com/operational-efficiency/artificial-intelligence/barriers-to-ai-adoption/ (accessed 6.26.25).

Eelbode, T., Bertels, J., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., Blaschko, M.B., 2020. Optimization for Medical Image Segmentation: Theory and Practice When Evaluating With Dice Score or Jaccard Index. IEEE Trans. Med. Imaging 39, 3679–3690. https://doi.org/10.1109/TMI.2020.3002417

El Jurdi, R., Petitjean, C., Honeine, P., Cheplygina, V., Abdallah, F., 2021. High-level prior-based loss functions for medical image segmentation:

A survey. Computer Vision and Image Understanding 210, 103248. https://doi.org/10.1016/J.CVIU.2021.103248

Elliott, J.W., Olbina, S., 2023. Benefits and Obstacles to the Adoption of Reality Capture Technologies in the U.S. Commercial and Infrastructure Construction Sectors. Buildings 13. https://doi.org/10.3390/buildings13030576

Farahmand, K., Vazaios, I., Diederichs, M.S., Vlachopoulos, N., 2018. Investigating the scale-dependency of the geometrical and mechanical properties of a moderately jointed rock using a synthetic rock mass (SRM) approach. Comput. Geotech. 95, 162–179. https://doi.org/10.1016/J.COMPGEO.2017.10.002

FARO, 2022. Faro Focus Premium.

Fehér, K., Török, Á., 2022. Detecting short-term weathering of stone monuments by 3D laser scanning: lithology, wall orientation, material loss. J. Cult. Herit. 58, 245–255. https://doi.org/10.1016/J.CULHER.2022.10.012

Fekete, S., Diederichs, M., Lato, M., 2010. Geotechnical and operational applications for 3-dimensional laser scanning in drill and blast tunnels. https://doi.org/10.1016/j.tust.2010.04.008

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. Tunnelling and Underground Space Technology 136, 105107. https://doi.org/10.1016/J.TUST.2023.105107

Feroz, S., Dabous, S.A., 2021. Uav-based remote sensing applications for bridge condition assessment. Remote Sens. (Basel). https://doi.org/10.3390/rs13091809

Fly4Future Ltd., 2025. Drones For Inspection Of Highway Tunnels - Fly4Future [WWW Document]. URL https://fly4future.com/development-and-prototyping/drones-for-inspection-of-highway-tunnels/ (accessed 6.26.25).

Forster, A.M., Douglas, J., 2010. Condition survey objectivity and philosophy driven masonry repair: An increased probability for project divergence? Structural Survey 28, 384–407. https://doi.org/10.1108/02630801011089173/FULL/XML

Fugro, 2022. RILA 3.0 [WWW Document]. URL https://www.fugro.com/about-fugro/our-expertise/innovations/rila (accessed 2.7.22).

Gal, Y., Ghahramani, Z., 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.

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. https://doi.org/10.1177/1369433219849829 22, 2907–2921. https://doi.org/10.1177/1369433219849829

Gao, Y., Shi, Y., Zhang, J., Ling, T., Lu, J., Zheng, Y., Yu, Q., Qi, L., 2022. Inconsistency-aware uncertainty estimation for semi-supervised medical image segmentation. IEEE Trans. Med. Imaging 41, 608–620. https://doi.org/10.1109/TMI.2021.3117888

Garrido, G.G., Sattar, T.P., 2021. An Autonomous Wall-Climbing Robot for Inspection of Reinforced Concrete Structures: SIRCAUR. Journal of Artificial Intelligence and Technology 1, 188–196. https://doi.org/10.37965/JAIT.2021.0016

Geng, Q., Ye, Y., Wang, X., 2022. Identifying void defects behind Tunnel composite lining based on transient electromagnetic radar method. NDT & E International 125, 102562. https://doi.org/10.1016/J.NDTEINT.2021.102562

GeoSLAM, 2022. ZEB Go Handheld 3D Scanner: Laser Scanning for Everyone [WWW Document]. URL https://geoslam.com/solutions/zeb-go/ (accessed 8.7.22).

Gibbons, S., Heblich, S., Pinchbeck, E.W., 2024. The spatial impacts of a massive rail disinvestment program: The Beeching Axe. J. Urban Econ. 143, 103691. https://doi.org/10.1016/J.JUE.2024.103691

Gilbert, M., Melbourne, C., 1995. The Behaviour of Multiring Brickwork Arch Bridges. The Structural Engineer 73.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 580–587. https://doi.org/10.48550/arxiv.1311.2524

Girshick, R., Microsoft Research, 2015. Fast R-CNN, in: IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/ICCV.2015.169

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

Granshaw, S.I., 2018. Structure from motion: origins and originality. Photogrammetric Record 33, 6–10. https://doi.org/10.1111/PHOR.12237

Grant, D., 1978. REPORT ON CPR - BROCKVILLE TUNNEL. Brockville.

Gray, R.M., Neuhoff, D.L., 1998. Quantization. IEEE Trans. Inf. Theory 44, 2325–2384. https://doi.org/10.1109/18.720541

Grilli, E., Özdemir, E., Remondino, F., 2019. Application of machine and deep learning strategies for the classification of heritage point clouds. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 42, 447–454. https://doi.org/10.5194/ISPRS-ARCHIVES-XLII-4-W18-447-2019

Gripper, C., 1879. Railway Tunnelling in Heavy Ground . E. & F. N. SPON.

Groves, G.L., 1943. TUNNEL LININGS, WITH SPECIAL REFERENCE TO A NEW FORM OF REINFORCED CONCRETE LINING. Journal of the Institution of Civil Engineers 20, 29–42. https://doi.org/10.1680/IJOTI.1943.13865

Guan, H., Liu, M., 2022. Domain Adaptation for Medical Image Analysis: A Survey. IEEE Trans. Biomed. Eng. 69, 1173–1185. https://doi.org/10.1109/TBME.2021.3117407

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

Haack, A., Schreyer, J., Jackel, G., 1995. Report to ITA Working Group on Maintenance and Repair of underground structures: State-of-the-art of non-destructive testing methods for determining the state of a tunnel lining. Tunnelling and Underground Space Technology 10, 413–431. https://doi.org/10.1016/0886-7798(95)00030-3

Hackel, T., Wegner, J.D., Schindler, K., 2016a. Contour detection in unstructured 3D point clouds, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Las Vegas.

Hackel, T., Wegner, J.D., Schindler, K., 2016b. FAST SEMANTIC SEGMENTATION OF 3D POINT CLOUDS WITH STRONGLY VARYING DENSITY, in: ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. pp. 177–184. https://doi.org/10.5194/isprsannals-III-3-177-2016

Halder, S., Afsari, K., 2023. Robots in Inspection and Monitoring of Buildings and Infrastructure: A Systematic Review. Applied Sciences 2023, Vol. 13, Page 2304 13, 2304. https://doi.org/10.3390/APP13042304

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

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. https://doi.org/10.1038/S41586-020-2649-2

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN. IEEE Trans. Pattern Anal. Mach. Intell. 42, 386–397. https://doi.org/10.48550/arxiv.1703.06870

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on

Computer Vision and Pattern Recognition 2016-December, 770–778. https://doi.org/10.1109/CVPR.2016.90

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

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors.

Höfle, B., Pfeifer, N., 2007. Correction of laser scanning intensity data: Data and model-driven approaches. ISPRS Journal of Photogrammetry and Remote Sensing 62, 415–433. https://doi.org/10.1016/J.ISPRSJPRS.2007.05.008

Holynski, A., Geraghty, D., Frahm, J.M., Sweeney, C., Szeliski, R., 2020. Reducing Drift in Structure from Motion Using Extended Features. Proceedings - 2020 International Conference on 3D Vision, 3DV 2020 51–60. https://doi.org/10.1109/3DV50981.2020.00015

Hsieh, Y.-A., Tsai, Y.J., 2020. Machine Learning for Crack Detection: Review and Model Performance Comparison. Journal of Computing in Civil Engineering 34, 04020038. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000918

Hu, C., Sapkota, B.B., Thomasson, J.A., Bagavathiannan, M. V., 2021. Influence of Image Quality and Light Consistency on the Performance of Convolutional Neural Networks for Weed Mapping. Remote Sensing 2021, Vol. 13, Page 2140 13, 2140. https://doi.org/10.3390/RS13112140

Hu, X., 2021. Structure-Aware Image Segmentation with Homotopy Warping. Adv. Neural Inf. Process. Syst. 35.

Hu, X., Fuxin, L., Samaras, D., Chen, C., 2019. Topology-Preserving Deep Image Segmentation. Adv. Neural Inf. Process. Syst. 32.

Huang, H., Cai, Y., Zhang, C., Lu, Y., Hammad, A., Fan, L., 2024. Crack detection of masonry structure based on thermal and visible image fusion and semantic segmentation. Autom. Constr. 158, 105213. https://doi.org/10.1016/J.AUTCON.2023.105213

Huang, H., Cheng, W., Zhou, M., Chen, J., Zhao, S., 2020a. Towards automated 3d inspection of water leakages in shield tunnel linings using mobile laser scanning data. Sensors (Switzerland) 20, 1–23. https://doi.org/10.3390/S20226669

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

Hussain Rather, I., Sushil Kumar, ·, Amir, ·, Gandomi, H., 2024. Breaking the data barrier: a review of deep learning techniques for democratizing AI

with small datasets. Artificial Intelligence Review 2024 57:9 57, 1–61. https://doi.org/10.1007/S10462-024-10859-3

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

Ibrahim, Y., Nagy, B., Benedek, C., 2019. CNN-based watershed marker extraction for brick segmentation in masonry walls, in: ICIAR 2019: Image Analysis and Recognition. Springer Verlag, pp. 332–344. https://doi.org/10.1007/978-3-030-27202-9_30

Idjaton, K., Desquesnes, X., Treuillet, S., Brunetaud, X., 2021. Stone-by-Stone Segmentation for Monitoring Large Historical Monuments Using Deep Neural Networks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 12667 LNCS, 235–248. https://doi.org/10.1007/978-3-030-68787-8_17/FIGURES/7

Isensee, F., Wald, T., Ulrich, C., Baumgartner, M., Roy, S., Maier-Hein, K., Jäger, P.F., 2024. nnU-Net Revisited: A Call for Rigorous Validation in 3D Medical Image Segmentation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 15009 LNCS, 488–498. https://doi.org/10.1007/978-3-031-72114-4_47

Jadhav, M., Nichat, K., 2021. Life Cycle Cost Analysis of Tunnel. International Research Journal of Engineering and Technology.

Ji, A., Chew, A.W.Z., Xue, X., Zhang, L., 2022. An encoder-decoder deep learning method for multi-class object segmentation from 3D tunnel

point clouds. Autom. Constr. 137, 104187. https://doi.org/10.1016/J.AUTCON.2022.104187

Jing, Y., Sheil, B., Acikgoz, S., 2024. A lightweight Transformer-based neural network for large-scale masonry arch bridge point cloud segmentation. Computer-Aided Civil and Infrastructure Engineering 39, 2427–2438. https://doi.org/10.1111/MICE.13201

Jing, Y., Sheil, B., Acikgoz, S., 2022. Segmentation of large-scale masonry arch bridge point clouds with a synthetic simulator and the BridgeNet neural network. Autom. Constr. 142, 104459. https://doi.org/10.1016/J.AUTCON.2022.104459

Jordan, M.I., Mitchell, T.M., 2015. Machine learning: Trends, perspectives, and prospects. Science (1979). 349, 255–260. https://doi.org/10.1126/SCIENCE.AAA8415

Kaartinen, E., Dunphy, K., Sadhu, A., 2022. LiDAR-Based Structural Health Monitoring: Applications in Civil Infrastructure Systems. Sensors (Basel). 22, 4610. https://doi.org/10.3390/S22124610

Kajatin, R., Nalpantidis, L., 2021. Image Segmentation of Bricks in Masonry Wall Using a Fusion of Machine Learning Algorithms, in: ICPR 2021: Pattern Recognition. ICPR International Workshops and Challenges. Springer Science and Business Media Deutschland GmbH, pp. 446–461. https://doi.org/10.1007/978-3-030-68787-8_33/FIGURES/5

Kamrul Hasan, S.M., Linte, C.A., 2019. U-NetPlus: A Modified Encoder-Decoder U-Net Architecture for Semantic and Instance Segmentation of Surgical Instruments from Laparoscopic Images. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine

and Biology Society, EMBS 7205–7211. https://doi.org/10.1109/EMBC.2019.8856791

Kang, J., Chen, N., Li, M., Mao, S., Zhang, H., Fan, Y., Liu, H., 2023. A Point Cloud Segmentation Method for Dim and Cluttered Underground Tunnel Scenes Based on the Segment Anything Model. Remote Sensing 2024, Vol. 16, Page 97 16, 97. https://doi.org/10.3390/RS16010097

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. https://doi.org/10.1080/15583058.2023.2201576. https://doi.org/10.1080/15583058.2023.2201576

Keitaanniemi, A., Rönnholm, P., Kukko, A., Vaaja, M.T., 2023. Drift analysis and sectional post-processing of indoor simultaneous localization and mapping (SLAM)-based laser scanning data. Autom. Constr. 147, 104700. https://doi.org/10.1016/J.AUTCON.2022.104700

Kendall, A., Badrinarayanan, V., Cipolla, R., 2017. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. British Machine Vision Conference 2017, BMVC 2017. https://doi.org/10.5244/C.31.57

Kendall, A., Gal, Y., n.d. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?

Kennedy, C.J., Penman, M., Watkinson, D., Emmerson, N., Thickett, D., Bosché, F., Forster, A.M., Grau-Bové, J., Cassar, M., 2024. Beyond Heritage Science: A Review. Heritage 2024, Vol. 7, Pages 1510-1538 7, 1510–1538. https://doi.org/10.3390/HERITAGE7030073

Keshmiry, A., Hassani, S., Dackermann, U., Li, J., 2024. Assessment, repair, and retrofitting of masonry structures: A comprehensive review. Constr. Build. Mater. 442, 137380. https://doi.org/10.1016/J.CONBUILDMAT.2024.137380

Kheradmandi, N., Mehranfar, V., 2022. A critical review and comparative study on image segmentation-based techniques for pavement crack detection. Constr. Build. Mater. 321, 126162. https://doi.org/10.1016/J.CONBUILDMAT.2021.126162

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

Kiureghian, A. Der, Ditlevsen, O., 2009. Aleatory or epistemic? Does it matter? Structural Safety 31, 105–112. https://doi.org/10.1016/J.STRUSAFE.2008.06.020

Klaaßen, L., Steffen, B., 2023. Meta-analysis on necessary investment shifts to reach net zero pathways in Europe. Nat. Clim. Chang. 13, 58–66. https://doi.org/10.1038/S41558-022-01549-5;TECHMETA=141;SUBJMETA=559,682,689,694,703,704,706,844;KWRD=CLIMATE-CHANGE+MITIGATION,TECHNOLOGY

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. Advanced Engineering Informatics 29, 196–210. https://doi.org/10.1016/J.AEI.2015.01.008

Konishi, S., Kawakami, K., Taguchi, M., 2016. Inspection Method with Infrared Thermometry for Detect Void in Subway Tunnel Lining.

Procedia Eng. 165, 474–483. https://doi.org/10.1016/J.PROENG.2016.11.723

Kravitz, B., Mooney, M., Karlovsek, J., Danielson, I., Hedayat, A., 2019. Void detection in two-component annulus grout behind a pre-cast segmental tunnel liner using Ground Penetrating Radar. Tunnelling and Underground Space Technology 83, 381–392. https://doi.org/10.1016/J.TUST.2018.09.032

Küçükdemirci, M., Sarris, A., 2022. GPR Data Processing and Interpretation Based on Artificial Intelligence Approaches: Future Perspectives for Archaeological Prospection. Remote Sensing 2022, Vol. 14, Page 3377 14, 3377. https://doi.org/10.3390/RS14143377

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

Lambert, B., Forbes, F., Doyle, S., Dehaene, H., Dojat, M., 2024. Trustworthy clinical AI solutions: A unified review of uncertainty quantification in Deep Learning models for medical image analysis. Artif. Intell. Med. 150, 102830. https://doi.org/10.1016/J.ARTMED.2024.102830

Lattanzi, D., Asce, M., Miller, G., 2017. Review of Robotic Infrastructure Inspection Systems. Journal of Infrastructure Systems 23, 04017004. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000353

Laves, M.H., Ihler, S., Ortmaier, T., Kahrs, L.A., 2019. Quantifying the uncertainty of deep learning-based computer-aided diagnosis for patient safety. Current Directions in Biomedical Engineering 5, 223–226. https://doi.org/10.1515/cdbme-2019-0057

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

Lei, X., Dong, Y., Frangopol, D.M., 2023. Sustainable Life-Cycle Maintenance Policymaking for Network-Level Deteriorating Bridges with a Convolutional Autoencoder–Structured Reinforcement Learning Agent. Journal of Bridge Engineering 28, 04023063. https://doi.org/10.1061/JBENF2.BEENG-6159

Lei, X., Sun, M., Zhao, R., Wu, H., Zhou, Z., Dong, Y., Sun, L., 2024. Unsupervised Vision-Based Structural Anomaly Detection and Localization with Reverse Knowledge Distillation. Struct. Control Health Monit. 2024. https://doi.org/10.1155/2024/8933148

Li, C.H., Tam, P.K.S., 1998. An iterative algorithm for minimum cross entropy thresholding. Pattern Recognit. Lett. 19, 771–776. https://doi.org/10.1016/S0167-8655(98)00057-9

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. https://doi.org/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

Li, X., Lin, X., Zhu, H., Wang, X., Liu, Z., 2017. Condition assessment of shield tunnel using a new indicator: The tunnel serviceability index. Tunnelling and Underground Space Technology 67, 98–106. https://doi.org/10.1016/J.TUST.2017.05.007

Lin, Q., Chen, X., Chen, C., Garibaldi, J.M., 2023. A Novel Quality Control Algorithm for Medical Image Segmentation Based on Fuzzy Uncertainty. IEEE Transactions on Fuzzy Systems 31, 2532–2544. https://doi.org/10.1109/TFUZZ.2022.3228332

Liu, K.Q., Zhong, S.S., Zhao, K., Song, Y., 2023. Motion control and positioning system of multi-sensor tunnel defect inspection robot: from methodology to application. Sci. Rep. 13, 1–13. https://doi.org/10.1038/S41598-023-27402-Z;SUBJMETA=166,639,986,988;KWRD=CIVIL+ENGINEERING,MECHANICAL+ENGINEERING

Liu, W., Qi, Y., Guan, T., Liang, Y., 2025. Efficient and accurate 3D reconstruction in a featureless tunnel environment: a robust LiDAR-Inertial SLAM tightly coupled with wheel odometry. Journal of Civil Structural Health Monitoring 2025 15:8 15, 3575–3588. https://doi.org/10.1007/S13349-025-01005-W

Llanca, D., Breul, P., Bacconnet, C., 2017. Improving the diagnosis methodology for masonry tunnels. Tunnelling and Underground Space Technology 70, 55–64. https://doi.org/10.1016/J.TUST.2017.07.002

Llanca, D., Breul, P., Bacconnet, C., Sahli, M., 2013. Characterization of the masonry lining of an underground structure by geoendoscopy. Tunnelling and Underground Space Technology 38, 254–265. https://doi.org/10.1016/J.TUST.2013.07.009

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 07-12-June-2015, 431–440. https://doi.org/10.1109/CVPR.2015.7298965

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

Loupe360 [WWW Document], 2024. . https://loupe360.arup.com/.

Loupos, K., Amditis, A., Doulamis, A., Chrobocinski, P., Victores, J., Wietek, M., Panetsos, P., Roncaglia, A., Camarinopoulos, S., Kallidromitis, V., Bairaktaris, D., Komodakis, N., Lopez, R., 2016. Integrated ROBOTIC solution for tunnel structural evaluation and characterization - ROBO-SPECT EC Project. Transforming the Future of Infrastructure through Smarter Information - Proceedings of the International Conference on Smart Infrastructure and Construction, ICSIC 2016 309–314. https://doi.org/10.1680/TFITSI.61279.309

Loverdos, D., Sarhosis, V., 2023. Geometrical digital twins of masonry structures for documentation and structural assessment using machine learning. Eng. Struct. 275, 115256. https://doi.org/10.1016/J.ENGSTRUCT.2022.115256

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

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60, 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Luo, Z., Yang, W., Yuan, Y., Gou, R., Li, X., 2024. Semantic segmentation of agricultural images: A survey. Information Processing in Agriculture 11, 172–186. https://doi.org/10.1016/J.INPA.2023.02.001

Makarov, I., Koshevaya, E., Pechenina, A., Boyko, G., Starshinova, A., Kudlay, D., Makarova, T., Mitrofanova, L., 2025. A Comparative Analysis of SegFormer, FabE-Net and VGG-UNet Models for the Segmentation of Neural Structures on Histological Sections. Diagnostics 15, 2408. https://doi.org/10.3390/DIAGNOSTICS15182408/S1

Maninis, K.K., Caelles, S., Pont-Tuset, J., Van Gool, L., 2017. Deep Extreme Cut: From Extreme Points to Object Segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 616–625. https://doi.org/10.48550/arxiv.1711.09081

Martí-Henneberg, J., 2021. From State-Building to European Integration: The Role of the Railway Network in the Territorial Integration of Europe, 1850–2020. Soc. Sci. Hist. 45, 221–231. https://doi.org/10.1017/SSH.2021.7

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. https://doi.org/10.1109/IROS.2015.7353481

McCormick, N., 2010. NPL REPORT MAT 42 Alternative Methods for Railway Tunnel Examination - A Review and Recommendations.

McCormick, N.J., Kimkeran, S.A., Najim, A., Jonas, D., 2014. Assessing the condition of railway assets using difcam: Results from tunnel examinations. IET Conference Publications 2014. https://doi.org/10.1049/CP.2014.1002

McDonald, T., Robinson, M., Tian, G.Y., 2023. T-Vision: A hybrid subsurface radar inspection system for intelligent asset management of railway

tunnels. Transportation Research Procedia 72, 3466–3473. https://doi.org/10.1016/J.TRPRO.2023.11.768

McDonald, T., Robinson, M., Tian, G.Y., 2022. Developments in 3D Visualisation of the Rail Tunnel Subsurface for Inspection and Monitoring. Applied Sciences (Switzerland) 12, 11310. https://doi.org/10.3390/APP122211310/S1

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

McNabb, M., 2025. Drone Tunnel Inspection System - DRONELIFE [WWW Document]. dronelife. URL https://dronelife.com/2025/04/09/drones-go-underground-japans-toda-construction-uses-uavs-to-inspect-tunnel-faces/ (accessed 6.26.25).

Medina, R., Llamas, J., Gómez-García-Bermejo, J., Zalama, E., Segarra, M., 2017. Crack detection in concrete tunnels using a gabor filter invariant to rotation. Sensors (Switzerland) 17. https://doi.org/10.3390/s17071670

Membah, J., Asa, E., 2015. Estimating cost for transportation tunnel projects: A systematic literature review. International Journal of Construction Management 15, 196–218. https://doi.org/10.1080/15623599.2015.1067345;JOURNAL:JOURNAL:TJCM20;WGROUP:STRING:PUBLICATION

Mienye, I.D., Sun, Y., Wang, Z., 2019. Prediction performance of improved decision tree-based algorithms: a review. Procedia Manuf. 35, 698–703. https://doi.org/10.1016/J.PROMFG.2019.06.011

Milletari, F., Navab, N., Ahmadi, S.A., 2016. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. Proceedings -

2016 4th International Conference on 3D Vision, 3DV 2016 565–571. https://doi.org/10.1109/3DV.2016.79

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D., 2022. Image Segmentation Using Deep Learning: A Survey. IEEE Trans. Pattern Anal. Mach. Intell. 44, 3523–3542. https://doi.org/10.1109/TPAMI.2021.3059968

Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., Lucic, M., 2021. Revisiting the Calibration of Modern Neural Networks. Adv. Neural Inf. Process. Syst. 19, 15682–15694.

Minh Dang, L., Wang, H., Li, Y., Nguyen, L.Q., Nguyen, T.N., Song, H.K., Moon, H., 2022. Deep learning-based masonry crack segmentation and real-life crack length measurement. Constr. Build. Mater. 359, 129438. https://doi.org/10.1016/J.CONBUILDMAT.2022.129438

Mo, Y., Wu, Y., Yang, X., Liu, F., Liao, Y., 2022. Review the state-of-the-art technologies of semantic segmentation based on deep learning. Neurocomputing 493, 626–646. https://doi.org/10.1016/J.NEUCOM.2022.01.005

Mohan, A., Poobal, S., 2018. Crack detection using image processing: A critical review and analysis. Alexandria Engineering Journal 57, 787–798. https://doi.org/10.1016/J.AEJ.2017.01.020

Molesworth, G.L., 1886. Pocket-book of useful formulae & memoranda for civil and mechanical engineers.

Montero, R., Victores, J.G., Martínez, S., Jardón, A., Balaguer, C., 2015. Past, present and future of robotic tunnel inspection. Autom. Constr. 59, 99–112. https://doi.org/10.1016/J.AUTCON.2015.02.003

Moore, S., n.d. East Portal, https://historicengland.org.uk/listing/the-list/list-entry/1201949?section=comments-and-photos.

Mosinska, A., Marquez-Neila, P., Kozinski, M., Fua, P., 2017. Beyond the Pixel-Wise Loss for Topology-Aware Delineation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition 3136–3145. https://doi.org/10.1109/CVPR.2018.00331

Mouaddib, E.M., Pamart, A., Pierrot-Deseilligny, M., Girardeau-Montaut, D., 2024. 2D/3D data fusion for the comparative analysis of the vaults of Notre-Dame de Paris before and after the fire. J. Cult. Herit. 65, 221–231. https://doi.org/10.1016/J.CULHER.2023.06.012

Munawar, H.S., Hammad, A.W.A., Haddad, A., Soares, C.A.P., Waller, S.T., 2021. Image-Based Crack Detection Methods: A Review. Infrastructures (Basel). 6, 115. https://doi.org/10.3390/INFRASTRUCTURES6080115

Nair, T., Precup, D., Arnold, D.L., Arbel, T., 2020. Exploring uncertainty measures in deep networks for Multiple sclerosis lesion detection and segmentation. Med. Image Anal. 59, 101557. https://doi.org/10.1016/J.MEDIA.2019.101557

Neale, B.S., n.d. The Consequences of Poor Serviceability-and the Way Forward.

Nellen, P.M., Frank, A., Broennimann, R., Sennhauser, U.J., 2000. Optical fiber Bragg gratings for tunnel surveillance, in: Smart Structures and Materials 2000: Sensory Phenomena and Measurement Instrumentation for Smart Structures and Materials. SPIE, p. 263. https://doi.org/10.1117/12.388114

Network Rail, 2023. NR/CAT/STP/001: Catalogue of Network Rail Standards.

Network Rail, 2016. NR/L3/CIV/006/4C - Structures, Tunnels and Operational Property Examinations: Recording of Tunnel Condition Marking Index (TCMI).

Neves, M., Holt, C., McConnell, R., Marjane, G., 2018. Preservation of foundation systems on railway upgrades. Proceedings of the Institution of Civil Engineers: Engineering Sustainability 171, 81–89. https://doi.org/10.1680/jensu.16.00042

Newman, T.S., 1995. A Survey of Automated Visual Inspection. Computer Vision and Image Understanding 61, 231–262. https://doi.org/10.1006/CVIU.1995.1017

Nielsen, D., 2016. Tree Boosting With XGBoost Why Does XGBoost Win “Every” Machine Learning Competition? Norwegian University of Science and Technology.

Numba, 2024. Numba: A High Performance Python Compiler.

Office of Rail (UK), 2023. Annual efficiency and finance assessment of Network Rail 2023.

Omar, T., Nehdi, M.L., 2018. Condition Assessment of Reinforced Concrete Bridges: Current Practice and Research Challenges. Infrastructures 2018, Vol. 3, Page 36 3, 36. https://doi.org/10.3390/INFRASTRUCTURES3030036

Oses, N., Dornaika, F., Moujahid, A., 2014. Image-Based Delineation and Classification of Built Heritage Masonry. Remote Sensing 2014, Vol. 6, Pages 1863-1889 6, 1863–1889. https://doi.org/10.3390/RS6031863

Otsu, N., 1979. THRESHOLD SELECTION METHOD FROM GRAY-LEVEL HISTOGRAMS. IEEE Trans Syst Man Cybern SMC-9, 62–66. https://doi.org/10.1109/TSMC.1979.4310076

Papakonstantinou, I., Siwe, A.T., Madanat, S.M., 2025. How Closures of Bridges and Tunnels Affect Travel Time in New York City. Transp. Res. Rec. https://doi.org/10.1177/03611981251380580

Paraskevopoulou, C., Benardos, A., 2013. Assessing the construction cost of Greek transportation tunnel projects. Tunnelling and Underground Space Technology 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., Diederichs, M., 2018. Analysis of time-dependent deformation in tunnels using the Convergence-Confinement Method. Tunnelling and Underground Space Technology 71, 62–80. https://doi.org/10.1016/J.TUST.2017.07.001

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., Facebook, Z.D., Research, A.I., Lin, Z., Desmaison, A., Antiga, L., Srl, O., Lerer, A., 2017. Automatic differentiation in PyTorch.

Pavel Iakubovskii, 2019. Segmentation Models Pytorch. GitHub repository.

Pavoni, G., Giuliani, F., De Falco, A., Corsini, M., Ponchio, F., Callieri, M., Cignoni, P., 2022. On Assisting and Automatizing the Semantic Segmentation of Masonry Walls. ACM Journal on Computing and Cultural Heritage (JOCCH) 15, 1–17. https://doi.org/10.1145/3477400

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

Perry, J., 2012. Waterloo Tunnel [WWW Document]. URL http://www.forgottenrelics.org/tunnels/waterloo-tunnel/ (accessed 6.27.25).

Peters, W.H., Ranson, W.F., 1982. Digital Imaging Techniques In Experimental Stress Analysis. Optical Engineering 21, 427–431. https://doi.org/10.1117/12.7972925

Petroutsatou, K., Maravas, A., Saramourtsis, A., 2021. A life cycle model for estimating road tunnel cost. Tunnelling and Underground Space Technology 111, 103858. https://doi.org/10.1016/J.TUST.2021.103858

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

Pielawski, N., Wählby, C., 2020. Introducing Hann windows for reducing edge-effects in patch-based image segmentation. PLoS One 15, e0229839. https://doi.org/10.1371/JOURNAL.PONE.0229839

Pierdicca, R., Paolanti, M., Matrone, F., Martini, M., Morbidoni, C., Malinverni, E.S., Frontoni, E., Lingua, A.M., 2020. Point Cloud Semantic Segmentation Using a Deep Learning Framework for Cultural Heritage. Remote Sensing 12, 1005. https://doi.org/10.3390/RS12061005

Plus Tell-Tale - Avongard [WWW Document], 2025. . https://www.avongard.com/product/plus-tell-tale-2/.

Prewitt, J., 1970. Object enhancement and extraction. Picture processing and Psychopictorics 75–149.

Prewitt, J.M.S., Mendelsohn, M.L., 1966. THE ANALYSIS OF CELL IMAGES*. Ann. N. Y. Acad. Sci. 128, 1035–1053. https://doi.org/10.1111/J.1749-6632.1965.TB11715.X

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. Applied Intelligence 49, 2793–2806. https://doi.org/10.1007/S10489-018-01396-Y/TABLES/8

QGIS_3.38.2, 2025. . https://qgis.org/.

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). https://doi.org/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.

railwaydata.co.uk, 2024. RailwayData - Tunnels [WWW Document]. URL https://www.railwaydata.co.uk/tunnels/ (accessed 7.31.24).

Rainio, O., Teuho, J., Klén, R., 2024. Evaluation metrics and statistical tests for machine learning. Scientific Reports 2024 14:1 14, 1–14. https://doi.org/10.1038/s41598-024-56706-x

Rao, A.S., Nguyen, T., Palaniswami, M., Ngo, T., 2020. Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure: Struct. Health Monit. 20, 2124–2142. https://doi.org/10.1177/1475921720965445

Rathnakumar, R., Pang, Y., Liu, Y., 2023. Epistemic and aleatoric uncertainty quantification for crack detection using a Bayesian Boundary Aware

Convolutional Network. Reliab. Eng. Syst. Saf. 240. https://doi.org/10.1016/J.RESS.2023.109547

ReCap Pro | Autodesk [WWW Document], n.d. . https://www.autodesk.com/products/recap/free-trial?msockid=2611cf3466706ec52d9edb6d67576fb6.

Regona, M., Yigitcanlar, T., Xia, B., Li, R.Y.M., 2022. Opportunities and Adoption Challenges of AI in the Construction Industry: A PRISMA Review. Journal of Open Innovation: Technology, Market, and Complexity 8, 45. https://doi.org/10.3390/JOITMC8010045

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

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

Rezaie, A., Achanta, R., Godio, M., Beyer, K., 2020. Comparison of crack segmentation using digital image correlation measurements and deep learning. Constr. Build. Mater. 261, 120474. https://doi.org/10.1016/J.CONBUILDMAT.2020.120474

Richards, J.A., 1998. Inspection, maintenance and repair of tunnels: International lessons and practice. Tunnelling and Underground Space Technology 13, 369–375. https://doi.org/10.1016/S0886-7798(98)00079-0

Ridler, T.W., Calvard, S., 1978. PICTURE THRESHOLDING USING AN ITERATIVE SLECTION METHOD. IEEE Trans. Syst. Man Cybern. SMC-8, 630–632. https://doi.org/10.1109/TSMC.1978.4310039

Riisgaard, S., Blas, M.R., 2005. SLAM for Dummies - A Tutorial Approach to Simultaneous Localization and Mapping [WWW Document]. URL https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.208.6289 (accessed 3.8.22).

Riveiro, B., DeJong, M.J., Conde, B., 2016. Automated processing of large point clouds for structural health monitoring of masonry arch bridges. Autom. Constr. 72, 258–268. https://doi.org/10.1016/j.autcon.2016.02.009

Roberts, J., 1974. THE RAILWAY TUNNEL AT BROCKVILLE. Brockville.

Robinson, D.C., Chung, H., Ryan, K., 2016. Numerical investigation of a hovering micro rotor in close proximity to a ceiling plane. J. Fluids Struct. 66, 229–253. https://doi.org/10.1016/J.JFLUIDSTRUCTS.2016.08.001

ROGERS, A.E., 1901. THE LOCATION AND CONSTRUCTION OF RAILWAY TUNNELS, WITH PARTICULARS OF SOME PRECENT WORK. Minutes of the Proceedings of the Institution of Civil Engineers 146, 191–201. https://doi.org/10.1680/IMOTP.1901.18542

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.

Saif, W., 2022. Photogrammetry: A Brief Historical Overview.

Sajedi, S.O., Liang, X., 2020. Model Uncertainty Quantification for Reliable Deep Vision Structural Health Monitoring.

Salehi, S.S.M., Erdogmus, D., Gholipour, A., 2017. Tversky loss function for image segmentation using 3D fully convolutional deep networks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10541 LNCS, 379–387. https://doi.org/10.1007/978-3-319-67389-9_44/FIGURES/4

Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.R., 2021. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. Proceedings of the IEEE 109, 247–278. https://doi.org/10.1109/JPROC.2021.3060483

Sánchez-Aparicio, L.J., Del Pozo, S., Ramos, L.F., Arce, A., Fernandes, F.M., 2018. Heritage site preservation with combined radiometric and geometric analysis of TLS data. Autom. Constr. 85, 24–39. https://doi.org/10.1016/J.AUTCON.2017.09.023

Sánchez-Rodríguez, A., Riveiro, B., Soilán, M., González-deSantos, L.M., 2018. Automated detection and decomposition of railway tunnels from Mobile Laser Scanning Datasets. Autom. Constr. 96, 171–179. https://doi.org/10.1016/J.AUTCON.2018.09.014

Sandrone, F., Labiouse, V., 2011. Identification and analysis of Swiss National Road tunnels pathologies. Tunnelling and Underground Space Technology 26, 374–390. https://doi.org/10.1016/J.TUST.2010.11.008

Sarker, I.H., 2021. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science 2021 2:3 2, 1–21. https://doi.org/10.1007/S42979-021-00592-X

SBRI, 2019. SBRI competition guidance: innovation in automated tunnel examination - GOV.UK [WWW Document]. URL

https://www.gov.uk/government/publications/sbri-competition-guidance-innovation-in-automated-tunnel-examination (accessed 4.1.22).

Schmidhuber, J., 2022. Annotated History of Modern AI and Deep Learning.

Schuller, M.P., 2003. Nondestructive testing and damage assessment of masonry structures. Progress in Structural Engineering and Materials 5, 239–251. https://doi.org/10.1002/PSE.160

Seeböck, P., Orlando, J.I., Schlegl, T., Waldstein, S.M., Bogunović, H., Klimscha, S., Langs, G., Schmidt-Erfurth, U., 2019. Exploiting Epistemic Uncertainty of Anatomy Segmentation for Anomaly Detection in Retinal OCT. IEEE Trans. Med. Imaging 39, 87–98. https://doi.org/10.1109/TMI.2019.2919951

Seib, V., Lange, B., Wirtz, S., 2020. Mixing Real and Synthetic Data to Enhance Neural Network Training - A Review of Current Approaches. ArXiv.

Sezgin, M., Sankur, B., 2004. Survey over image thresholding techniques and quantitative performance evaluation. https://doi.org/10.1117/1.1631315 13, 146–165. https://doi.org/10.1117/1.1631315

Shah, D., 2023. Intersection over Union (IoU): Definition, Calculation, Code [WWW Document]. V7labs. URL https://www.v7labs.com/blog/intersection-over-union-guide (accessed 12.24.25).

Shen, Y., Wang, J., Puente, I., 2020. A Novel Baseline-Based Method to Detect Local Structural Changes in Masonry Walls Using Dense

Terrestrial Laser Scanning Point Clouds. IEEE Sens. J. 20, 6504–6515. https://doi.org/10.1109/JSEN.2020.2975011

Shen, Y., Wang, J., Wei, S., Zheng, D., Ferreira, V.G., 2019. Accurate extraction of brick shapes in masonry walls from dense terrestrial laser scanning point cloud. Measurement 146, 254–267. https://doi.org/10.1016/J.MEASUREMENT.2019.05.086

Sheppard, G., n.d. East Portal, https://commons.wikimedia.org/wiki/File:2021_at_Bradford_Tunnel_-_east_portal.JPG.

Shit, S., Paetzold, J.C., Sekuboyina, A., Ezhov, I., Unger, A., Zhylka, A., Pluim, J.P.W., Bauer, U., Menze, B.H., 2020. clDice -- A Novel Topology-Preserving Loss Function for Tubular Structure Segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 16555–16564. https://doi.org/10.1109/CVPR46437.2021.01629

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on Image Data Augmentation for Deep Learning. J. Big Data 6, 60. https://doi.org/10.1186/s40537-019-0197-0

Shou, W., Wang, X., Wang, J., Hou, L., Truijens, M., 2014. Integration of BIM and lean concepts to improve maintenance efficiency: A case study. Computing in Civil and Building Engineering - Proceedings of the 2014 International Conference on Computing in Civil and Building Engineering 373–380. https://doi.org/10.1061/9780784413616.047;CTYPE:STRING:BOOK

Simmons, J., 1991. Victorian railway. Thames and Hudson.

Simonyan, K., Zisserman, A., 2014a. Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.

Simonyan, K., Zisserman, A., 2014b. Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.

Singh, Y., Farrelly, C.M., Hathaway, Q.A., Leiner, T., Jagtap, J., Carlsson, G.E., Erickson, B.J., 2023. Topological data analysis in medical imaging: current state of the art. Insights Imaging 14, 1–10. https://doi.org/10.1186/S13244-023-01413-W/FIGURES/7

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. https://doi.org/10.3390/S23063189

Smith, J., Paraskevopoulou, C., 2025. Practical assessment of masonry tunnel joint segmentation using topological machine learning. Civil Engineering Design 7, 93–110. https://doi.org/10.1002/CEND.202400049

Smith, J., Paraskevopoulou, C., 2024. 3D Deep Learning for Segmentation of Masonry Tunnel Joints, in: SMAR 2024 – 7th International Conference on Smart Monitoring, Assessment and Rehabilitation of Civil Structures. Structural Integrity Procedia.

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. https://doi.org/10.1201/9781003348030-351

Smith, J., Paraskevopoulou, C., Cohn, A.G., Kromer, R., Bedi, A., Invernici, M., 2024. Automated masonry spalling severity segmentation in historic railway tunnels using deep learning and a block face plane fitting approach. Tunnelling and Underground Space Technology 153, 106043. https://doi.org/10.1016/J.TUST.2024.106043

Smith, J.M.W., Paraskevopoulou, C., 2025. Uncertainty Quantification to Assess the Generalisability of Automated Masonry Joint Segmentation Methods. Infrastructures (Basel). 10, 98. https://doi.org/10.3390/INFRASTRUCTURES10040098

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 of the Photogrammetry, Remote Sensing and Spatial Information Sciences 5, 281–288. https://doi.org/10.5194/ISPRS-ANNALS-V-2-2020-281-2020

Soleymani, A., Jahangir, H., Nehdi, M.L., 2023. Damage detection and monitoring in heritage masonry structures: Systematic review. Constr. Build. Mater. 397, 132402. https://doi.org/10.1016/J.CONBUILDMAT.2023.132402

Solla, M., Pérez-Gracia, V., Fontul, S., 2021. A Review of GPR Application on Transport Infrastructures: Troubleshooting and Best Practices. Remote Sens. (Basel). 13, 672. https://doi.org/10.3390/rs13040672

Son, H., Bosché, F., Kim, C., 2015. As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A

survey. Advanced Engineering Informatics 29, 172–183. https://doi.org/10.1016/J.AEI.2015.01.009

Sony, S., Dunphy, K., Sadhu, A., Capretz, M., 2021. A systematic review of convolutional neural network-based structural condition assessment techniques. Eng. Struct. 226, 111347. https://doi.org/10.1016/J.ENGSTRUCT.2020.111347

Sourget, T., Hasany, S.N., Mériaudeau, F., Petitjean, C., 2024. Can SegFormer be a True Competitor to U-Net for Medical Image Segmentation? Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 14122 LNCS, 111–118. https://doi.org/10.1007/978-3-031-48593-0_8/FIGURES/4

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

Stent, S., Gherardi, R., Stenger, B., Soga, K., Cipolla, R., 2016. Visual change detection on tunnel linings. Mach. Vis. Appl. 27, 319–330. https://doi.org/10.1007/S00138-014-0648-8

Stewart, J., 2025. Major transport projects governance and assurance review. London.

Stover, J.F.., Carnes, M.C.., 1999. The Routledge historical atlas of the American railroads. Routledge.

Strauss, A., Bien, J., Neuner, H., Harmening, C., Seywald, C., Österreicher, M., Voit, K., Pistone, E., Spyridis, P., Bergmeister, K., 2020. Sensing and monitoring in tunnels testing and monitoring methods for the

assessment of tunnels. Structural Concrete 21, 1356–1376. https://doi.org/10.1002/SUCO.201900444

STUDENT, 1908. THE PROBABLE ERROR OF A MEAN. Biometrika 6, 1–25. https://doi.org/10.1093/BIOMET/6.1.1

Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015. Multi-View Convolutional Neural Networks for 3D Shape Recognition.

Suchocki, C., 2020. Comparison of Time-of-Flight and Phase-Shift TLS Intensity Data for the Diagnostics Measurements of Buildings. Materials 13. https://doi.org/10.3390/MA13020353

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. https://doi.org/10.1007/978-3-319-67558-9_28

Sun, C., Shrivastava, A., Singh, S., Gupta, A., 2017. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. Proceedings of the IEEE International Conference on Computer Vision 2017-October, 843–852. https://doi.org/10.1109/ICCV.2017.97

Szeliski, R., 2022. Computer Vision, Texts in Computer Science. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-34372-9

Tang, S., Chen, W., Wang, W., Li, X., Darwish, W., Li, W., Huang, Z., Hu, H., Guo, R., 2018. Geometric integration of hybrid correspondences for RGB-D unidirectional tracking. Sensors (Switzerland) 18. https://doi.org/10.3390/S18051385

Thakker, D., Dimitrova, V., Cohn, A.G., Valdes, J., 2015. PADTUN - Using Semantic Technologies in Tunnel Diagnosis and Maintenance Domain. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9088, 683–698. https://doi.org/10.1007/978-3-319-18818-8_42

The Railway Year Book for 1915, 1915. . The Railway Publishing Co.

Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L., 2019. KPConv: Flexible and deformable convolution for point clouds. Proceedings of the IEEE International Conference on Computer Vision 6410–6419. https://doi.org/10.1109/ICCV.2019.00651

Thompson, L.F., Stowell, J.P., Fargher, S.J., Steer, C.A., Loughney, K.L., O'Sullivan, E.M., Gluyas, J.G., Blaney, S.W., Pidcock, R.J., 2020. Muon tomography for railway tunnel imaging. Phys. Rev. Res. 2. https://doi.org/10.1103/PHYSREVRESEARCH.2.023017

Tian, Y., Ma, S., 2018. An Analysis about the Problem of SLAM in Tunnel-like Environment with Serpentine Locomotion and Its Solution. 2018 IEEE International Conference on Robotics and Biomimetics, ROBIO 2018 1626–1631. https://doi.org/10.1109/ROBIO.2018.8665070

Tichý, T., Brož, J., Bělinová, Z., Pirník, R., 2021. Analysis of Predictive Maintenance for Tunnel Systems. Sustainability 2021, Vol. 13, Page 3977 13, 3977. https://doi.org/10.3390/SU13073977

Tokui, S., 2024. Cupy : NumPy & SciPy for GPU.

Trafikverket, 2017. REPORT 6A, Test of SLAM in a tunnel project - Evaluation of SLAM used for mobile mapping positioning.

Tukey, J.W., 1949. Comparing Individual Means in the Analysis of Variance. Biometrics 5, 99. https://doi.org/10.2307/3001913

Tyler, H.W., 1867. ON THE WORKING OF SHARP GRADIENTS AND SHARP CURVES ON RAILWAYS. https://doi.org/10.1680/imotp.1867.23164 26, 310–324. https://doi.org/10.1680/IMOTP.1867.23164

Tympakianaki, A., Koutsopoulos, H.N., Jenelius, E., Cebecauer, M., 2018. Impact analysis of transport network disruptions using multimodal data: A case study for tunnel closures in Stockholm. Case Stud. Transp. Policy 6, 179–189. https://doi.org/10.1016/J.CSTP.2018.05.003

UK Power Networks, 2023. UK Power Networks Ltd. [WWW Document]. URL https://www.ukpowernetworks.co.uk/news/robot-dog (accessed 6.26.25).

Ukai, M., 2007. Advanced Inspection System of Tunnel Wall Deformation using Image Processing. Quarterly Report of RTRI 48, 94–98. https://doi.org/10.2219/RTRIQR.48.94

Ukai, M., Miyamoto, T., Sasama, H., 1996. Development of inspection system of railway facilities using continuous scan image. COMPUTERS IN RAILWAYS V 1, 61–70.

Valdenegro-Toro, M., Saromo, D., 2022. A Deeper Look into Aleatoric and Epistemic Uncertainty Disentanglement. https://doi.org/10.52591/lxai202206244

Valero, Enrique, Bosché, F., Forster, A., 2018. Automatic segmentation of 3D point clouds of rubble masonry walls, and its application to building surveying, repair and maintenance. Autom. Constr. 96, 29–39. https://doi.org/10.1016/J.AUTCON.2018.08.018

Valero, E., Forster, A., Bosché, F., Renier, C., Hyslop, E., Wilson, L., 2018. High level-of-detail BIM and machine learning for automated masonry

wall defect surveying. ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things. https://doi.org/10.22260/ISARC2018/0101

Valero, E., Forster, A., Bosché, F., Wilson, L., Leslie, A., 2016. COMPARISON OF 3D REALITY CAPTURE TECHNOLOGIES FOR THE SURVEY OF STONE WALLS. https://doi.org/10.4995/ARQUEOLOGICA8.2016.2582

Van Der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T., 2014. Scikit-image: Image processing in python. PeerJ. https://doi.org/10.7717/peerj.453

van Zyl, C., Ye, X., Naidoo, R., 2024. Harnessing eXplainable artificial intelligence for feature selection in time series energy forecasting: A comparative analysis of Grad-CAM and SHAP. Appl. Energy 353, 122079. https://doi.org/10.1016/j.apenergy.2023.122079

Vandenabeele, L., Loverdos, D., Pfister, M., Sarhosis, V., 2024. Deep Learning for the Segmentation of Large-Scale Surveys of Historic Masonry: A New Tool for Building Archaeology Applied at the Basilica of St Anthony in Padua. International Journal of Architectural Heritage 18, 1749–1761. https://doi.org/10.1080/15583058.2023.2260771

Vassilev, H., Laska, M., Blankenbach, J., 2024. Uncertainty-aware point cloud segmentation for infrastructure projects using Bayesian deep learning. Autom. Constr. 164, 105419. https://doi.org/10.1016/J.AUTCON.2024.105419

Vasu, P.K.A., Gabriel, J., Zhu, J., Tuzel, O., Ranjan, A., 2022. MobileOne: An Improved One millisecond Mobile Backbone. Proceedings of the

IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2023-June, 7907–7917. https://doi.org/10.1109/CVPR52729.2023.00764

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention Is All You Need 1.

Vazaios, I., Vlachopoulos, N., Diederichs, M.S., 2017. Integration of Lidar-Based Structural Input and Discrete Fracture Network Generation for Underground Applications. Geotechnical and Geological Engineering 35, 2227–2251. https://doi.org/10.1007/S10706-017-0240-X/FIGURES/20

Vong, C.H., Ravitharan, R., Reichl, P., Chevin, J., Chung, H., 2018. Small Scale Unmanned Aerial System (UAS) for Railway Culvert and Tunnel Inspection. ICRT 2017: Railway Development, Operations, and Maintenance - Proceedings of the 1st International Conference on Rail Transportation 2017 2017-July, 1024–1032. https://doi.org/10.1061/9780784481257.102

Vora, S., Lang, A.H., Helou, B., Beijbom, O., 2020. PointPainting: Sequential Fusion for 3D Object Detection, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/CVPR42600.2020.00466

Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E., 2018. Deep Learning for Computer Vision: A Brief Review. Comput. Intell. Neurosci. 2018, 7068349. https://doi.org/10.1155/2018/7068349

Vuola, A.O., Akram, S.U., Kannala, J., 2019. Mask-RCNN and u-net ensembled for nuclei segmentation. Proceedings - International

Symposium on Biomedical Imaging 2019-April, 208–212. https://doi.org/10.1109/ISBI.2019.8759574

Waite, G., Harris, D., 2012. Haddon Tunnel [WWW Document]. URL http://www.forgottenrelics.org/tunnels/haddon-tunnel/ (accessed 6.27.25).

Wan, S., Guan, S., Tang, Y., 2023. Advancing Bridge Structural Health Monitoring: Insights into Knowledge-Driven and Data-Driven Approaches. Journal of Data Science and Intelligent Systems 2, 129–140. https://doi.org/10.47852/bonviewjdsis3202964

Wang, G., Li, W., Aertsen, M., Deprest, J., Ourselin, S., Vercauteren, T., 2019. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. Neurocomputing 338, 34–45. https://doi.org/10.1016/J.NEUCOM.2019.01.103

Wang, N., Zhao, Q., Li, S., Zhao, X., Zhao, P., 2018. Damage Classification for Masonry Historic Structures Using Convolutional Neural Networks Based on Still Images. Computer-Aided Civil and Infrastructure Engineering 33, 1073–1089. https://doi.org/10.1111/MICE.12411

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

Wang, P., Shi, G., 2024. Image segmentation of tunnel water leakage defects in complex environments using an improved Unet model. Sci. Rep. 14. https://doi.org/10.1038/s41598-024-75723-4

Wasserman, L., 2018. Topological Data Analysis. Annu. Rev. Stat. Appl. 5, 501–532. https://doi.org/10.1146/ANNUREV-STATISTICS-031017-100045/CITE/REFWORKS

Wei, X., Yu, R., Sun, J., 2020. View-GCN: View-Based Graph Convolutional Network for 3D Shape Analysis.

Weise, M., Böhms, | Michel, Allaix, D., Sánchez-Rodríguez, A., Rigotti, M., 2023. Importance of Digitalization and Standardization for Bridge and Tunnel Monitoring and Predictive Maintenance. ce/papers 6, 592–599. https://doi.org/10.1002/CEPA.2144

Westoby, M.J., Brasington, J., Glasser, N.F., Hambrey, M.J., Reynolds, J.M., 2012. 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. Geomorphology 179, 300–314. https://doi.org/10.1016/J.GEOMORPH.2012.08.021

Wickstrøm, K., Kampffmeyer, M., Jenssen, R., 2020. Uncertainty and interpretability in convolutional neural networks for semantic segmentation of colorectal polyps. Med. Image Anal. 60, 101619. https://doi.org/10.1016/J.MEDIA.2019.101619

Wu, H., Ao, X., Chen, Z., Liu, C., Xu, Z., Yu, P., 2019. Concrete Spalling Detection for Metro Tunnel from Point Cloud Based on Roughness Descriptor. J. Sens. 2019. https://doi.org/10.1155/2019/8574750

Wu, H., Zhu, Q., Guo, Y., Zheng, W., Zhang, L., Wang, Q., Zhou, R., Ding, Y., Wang, W., Pirasteh, S., Liu, M., 2022. Multi-level voxel representations for digital twin models of tunnel geological environment. International Journal of Applied Earth Observation and Geoinformation 112, 102887. https://doi.org/10.1016/J.JAG.2022.102887

Wu, X., Sahoo, D., Hoi, S.C.H., 2020. Recent advances in deep learning for object detection. Neurocomputing 396, 39–64. https://doi.org/10.1016/J.NEUCOM.2020.01.085

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., 2021. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. Adv. Neural Inf. Process. Syst. 15, 12077–12090.

Xie, Y., Tian, J., Zhu, X.X., 2019. Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation. IEEE Geosci. Remote Sens. Mag. 8, 38–59. https://doi.org/10.1109/mgrs.2019.2937630

Xu, H., He, H., Zhang, Y., Ma, L., Li, J., 2023. A comparative study of loss functions for road segmentation in remotely sensed road datasets. International Journal of Applied Earth Observation and Geoinformation 116, 103159. https://doi.org/10.1016/J.JAG.2022.103159

Xu, X., Yang, H., 2020. Vision Measurement of Tunnel Structures with Robust Modelling and Deep Learning Algorithms. Sensors 20, 4945. https://doi.org/10.3390/s20174945

Xu, X., Zhao, M., Shi, P., Ren, R., He, X., Wei, X., Yang, H., 2022. Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN. Sensors (Basel). 22. https://doi.org/10.3390/S22031215

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

Yang, Z., Wang, L., 2019. Learning Relationships for Multi-View 3D Object Recognition.

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

Yen, J.C., Chang, F.J., Chang, S., 1995. A New Criterion for Automatic Multilevel Thresholding. IEEE Transactions on Image Processing 4, 370–378. https://doi.org/10.1109/83.366472

Yoon, J.S., Sagong, M., Lee, J.S., Lee, K. sung, 2009. Feature extraction of a concrete tunnel liner from 3D laser scanning data. NDT & E International 42, 97–105. https://doi.org/10.1016/J.NDTEINT.2008.10.001

Yu, P., Wu, H., Liu, C., Xu, Z., 2018. Water leakage diagnosis in metro tunnels by intergration of laser point cloud and infrared thermal imaging. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 42, 2167–2171. https://doi.org/10.5194/ISPRS-ARCHIVES-XLII-3-2167-2018,

Yu, S.N., Jang, J.H., Han, C.S., 2007. Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. Autom. Constr. 16, 255–261. https://doi.org/10.1016/J.AUTCON.2006.05.003

Yuan, L., Guo, J., Wang, Q., 2020. Automatic classification of common building materials from 3D terrestrial laser scan data. Autom. Constr. 110, 103017. https://doi.org/10.1016/J.AUTCON.2019.103017

Yuan, X., Shi, J., Gu, L., 2021. A review of deep learning methods for semantic segmentation of remote sensing imagery. Expert Syst. Appl. 169, 114417. https://doi.org/10.1016/J.ESWA.2020.114417

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

Zhang, J., Li, Z., Pang, J., Yang, D., Yang, Z., Chen, Z., Long, G., Du, P., Huang, L., Song, T., Li, K., Cui, M., Bao, F., Liu, F., Wang, Y., Liu, D., Ouyang, X., Han, R., 2025. Muon absorption imaging for the density structure of metro tunnel overburden. Nucl. Instrum. Methods Phys. Res. A 1073, 170307. https://doi.org/10.1016/J.NIMA.2025.170307

Zhang, J., Lin, X., 2016. Advances in fusion of optical imagery and LiDAR point cloud applied to photogrammetry and remote sensing. Int. J. Image Data Fusion 8, 1–31. https://doi.org/10.1080/19479832.2016.1160960

Zhang, Y., Tino, P., Leonardis, A., Tang, K., 2021. A Survey on Neural Network Interpretability. IEEE Trans. Emerg. Top. Comput. Intell. 5, 726–742. https://doi.org/10.1109/TETCI.2021.3100641

Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X., 2019. Object Detection with Deep Learning: A Review. IEEE Trans. Neural Netw. Learn. Syst. 30, 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865

Zheng, Z., Oda, M., Misawa, K., Mori, K., 2022. Taking full advantage of uncertainty estimation: An uncertainty-assisted two-stage pipeline for multi-organ segmentation. Progress in Biomedical Optics and Imaging

- Proceedings of SPIE 12033, 120333M. https://doi.org/10.1117/12.2612535

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, Q.-Y., Park, J., Koltun, V., 2018. Open3D: A Modern Library for 3D Data Processing.

Zhou, S., Canchila, C., Song, W., 2023. Deep learning-based crack segmentation for civil infrastructure: data types, architectures, and benchmarked performance. Autom. Constr. 146, 104678. https://doi.org/10.1016/J.AUTCON.2022.104678

Zhou, Y., Tuzel, O., 2017. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 4490–4499. https://doi.org/10.48550/arxiv.1711.06396

Zhou, Z., Siddiquee, R., Tajbakhsh, N., Liang, J., 2018. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. https://doi.org/10.1007/978-3-030-00889-5_1

Zia, A., Khamis, A., Nichols, J., Tayab, U.B., Hayder, Z., Rolland, V., Stone, E., Petersson, L., 2024. Topological deep learning: a review of an emerging paradigm. Artif. Intell. Rev. 57, 1–24. https://doi.org/10.1007/S10462-024-10710-9/FIGURES/C

Zonouz, E.F., Pan, X., Hsu, Y.-C., Yang, T., 2023. 3D vision-based structural masonry damage detection.

APPENDIX A1– Machine Learning for Computer vision

Machine learning algorithms are used to automatically establish trends in large quantities of data with many parameters and determine the probabilities of different conclusions about the data. In the field of condition assessment, the task is typically to replicate the conclusions of human inspectors and assessors about a structure. There are many different methods; An outline can be found in (Sarker, 2021). The most common style of machine learning involves training and optimising algorithm parameters on a sample of data based on a known conclusion. This is known as supervised learning. A range of data scenarios need to be used for training to ensure that the algorithm will be generalised to the range of real-world scenarios that the algorithm will be applied to. In addition, the algorithm needs to be trained enough to properly characterise the input data, while avoiding over fitting. If overfitting occurs, then the model is simply learning the exact features of the training data, rather than transferable knowledge about the nature of the data. This reduces the accuracy of the algorithm when applied to previously unseen data. In general, more training data that is varied and representative of likely situations will result in a more effective model.

Machine learning involving visual or spatial input data is a part of the field of computer vision. Prior to convolutional neural networks, computer vision with machine learning involved extracting explicit features from images using existing algorithms and then classifying these features using machine learning. Also known as deep learning, neural networks are a machine learning method consisting of layers of interconnected neurons, each containing a simple equation (Lecun et al., 2015). The output of each neuron

consists of the summation of each input multiplied by a weight. Backpropagation allows the equation parameters to be optimised for transforming data from the input to a desired output, adjusting the weights given to each neuron input. 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, 2014a). CNNs have been applied extensively to computer vision. Using masonry wall spalling as an example, the outputs of various common computer vision tasks are visualised in Figure A1. The tasks are ordered roughly in order of difficulty.

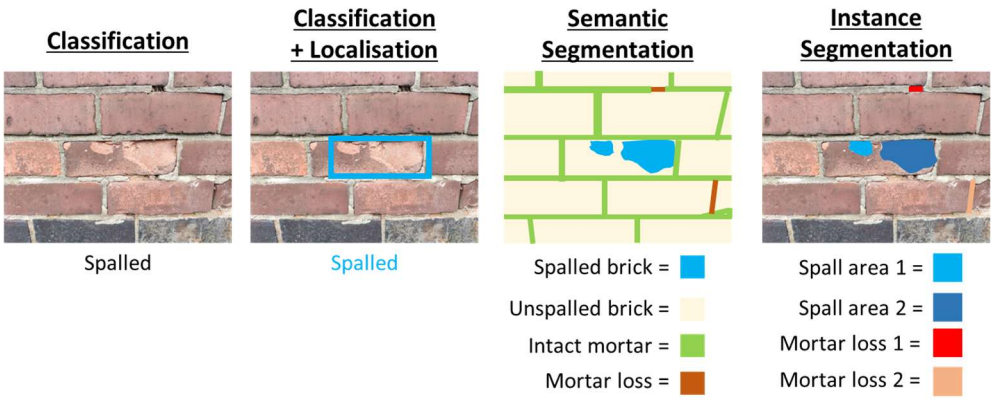


Figure A1: Outputs of computer vision tasks applied to masonry spalling detection.

Classification is the task of determining an image class from a list of possible classes. In Figure 1, there may be three classes - Spalled, Cracked or Undamaged. In this case, the image has been classified as being more likely to contain spalled masonry, than cracked or undamaged masonry. Convolutional neural networks work well at this problem and apply image convolutions, to create feature maps in each layer. The image convolutions learn to identify image features and are combined with a fully connected neural network for generating a conclusion about the image from those features. The nature of the convolutions is learnt during the training process. From the top to bottom layers of the network, the image is transformed to increasingly larger numbers of lower resolution feature maps, each highlighting a significant feature of the image. The image size is reduced using pooling layers and activation functions add non linearity to the data to enable more complex relationships to be learnt. In effect, this creates a list of binary image features that act as a checklist describing the image in the final layer. This list is compared against the common features of each class to select the optimal image classification. A diagram visualising a typical CNN design for image classification is shown in Figure A2.

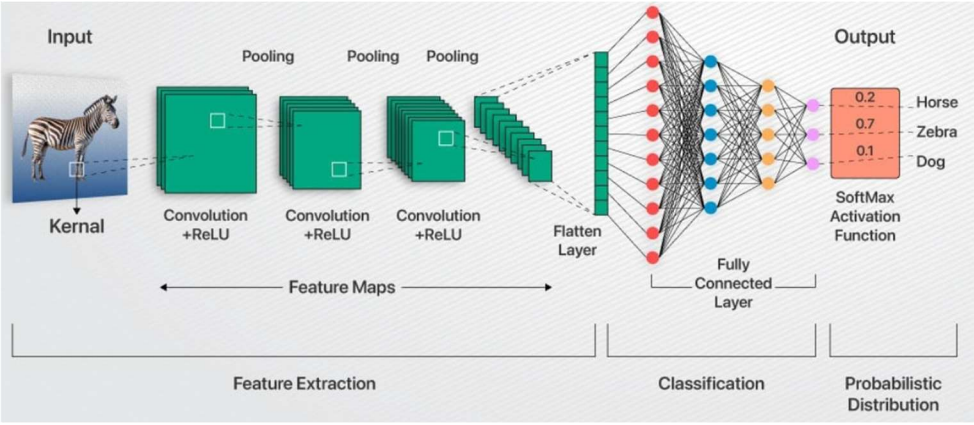


Figure A2: Typical CNN architecture, from (AnalytixLabs, 2024)

Classification can be expanded to include localisation. This involves placing a bounding box over the object identified in the classification of an image. The first efficient method of achieving this this was the R-CNN by (Girshick et al., 2013) (since improved upon by the fast R-CNN by (Girshick and Microsoft Research, 2015) and faster R-CNN by (Ren et al., 2017)). Localisation typically involves comparing the classification output from different cropped parts of the image to determine the part containing the object of interest. R-CNN, however, reduces the amount of region proposals required, by using a selective search algorithm. This uses a greedy algorithm to recursively combine similar regions into larger ones. (Wu et al., 2020; Zhao et al., 2019) have conducted reviews of different object detection algorithms involving object localisation.

Deep learning methods have been extensively applied to the field of semantic segmentation. Semantic segmentation involves classifying individual pixels as being part of a class of object. In Figure A1, a pixel could either be in an area of Brick Spalling, Unspalled Brick, Intact Mortar or Mortar loss. (Minaee et al., 2022) recently conducted a review of the field. One of the most influential algorithms for semantic segmentation is the U-Net developed by (Ronneberger et al., 2015) and shown in Figure A3. Although many newer designs of semantic segmentation algorithms have been proposed, most adapt or expand upon a core U-Net style architecture. Studies have shown that a well trained U-Net style model is often still the most effective deep learning model for many semantic segmentation tasks (Isensee et al., 2024).

Instance segmentation is a more challenging task and combines instance detection with semantic segmentation. It involves labelling different instances of the same class. In the example shown in Figure 1, the network needs to

distinguish that there are 2 different areas of spalling in the image as well as two areas of damaged mortar. Each Pixel in the image is labelled with a class representing each instance of objects in the image. A common network for instance segmentation is the mask R-CNN (He et al., 2017), which expands upon the R-CNN object detection method to segment the target object within each bounding box. The U-Net can also be adapted to deal with different instances of objects (Kamrul Hasan and Linte, 2019). (Vuola et al., 2019) showed that combining the results of a U-Net and a Mask R-CNN into an ensemble method produces better instance segmentation results than each model individually, as each algorithm makes different mistakes. Semi-automated segmentation methods, for example Deep Extreme Cut by (Maninis et al., 2017), have also been developed to aid manual segmentation. These can operate with only general training and limited to no training required on the target problem. A human operator only needs to select certain points on a target object, then the model will automatically segment the target object. This is substantially quicker than manually drawing an object outline and can be easier than training a full segmentation model for a specific problem.