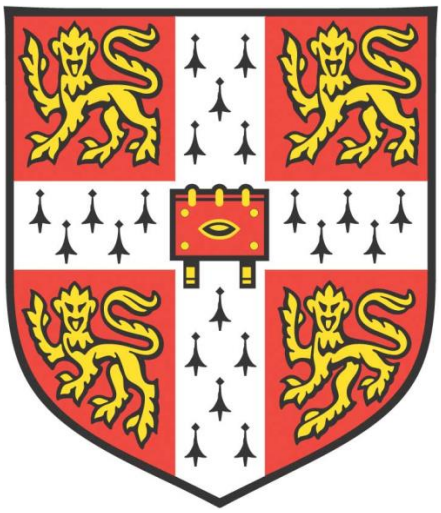
AUTOMATED IMAGE-BASED INSPECTION OF MASONRY ARCH BRIDGES



Daniel Brackenbury

St John's College

Department of Engineering

University of Cambridge

This thesis is submitted for the degree of Doctor of Philosophy

January 2022

DECLARATION

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee

Daniel Brackenbury

Cambridge

ABSTRACT

Title: Automated image-based inspection of masonry arch bridges

Author: *Daniel Brackenbury*

Masonry arch bridges have proven to be durable and underpin much of the world's transport infrastructure. However, they are an aging asset, and their effective management is challenging. Existing management focuses heavily on manual inspection, which has been shown as subjective and often leads to incomplete records regarding damage and retrofit of the structure.

Improved data collection and processing technologies now provide the opportunity to create a digital record of the entire bridge surface. However, if this digital record is still inspected manually, subjectivity will persist. While considerable progress has been made in automated image-based defect detection of concrete and asphalt infrastructure, relatively little progress has been made for masonry.

In this context, a new framework, or pipeline, for automated inspection of masonry arch bridges is presented: from the data capture phase all the way to the diagnosis of the underlying problems with the bridge. After presenting this framework, the focus of this work is on the automated detection of defects within realistic, deteriorated masonry surfaces that are typical of historic masonry arch bridges. This has involved the creation of a pixel-wise annotated dataset of masonry arch bridge surfaces for both different defect classes and mortar joints. This dataset is believed to be unparalleled in both scope and scale, compared to other works in the literature and therefore serves as an invaluable tool for future research in this area.

Methodologies for mapping the mortar joints on masonry surfaces have been examined, including a comparison of a hardcoded deterministic algorithm based on pattern detection, with a semantic deep learning model. Both methods performed well; the deep learning model was more robust to noisier image conditions. The effect of mortar joints on automated defect detection procedures was then investigated, as these joints are one of the key distractors and differentiators between masonry and concrete/asphalt surfaces. Whilst simple models based on edge detection were unable to differentiate defects from the masonry surface without prior segmentation of mortar, a more robust classifier based on a Convolutional Neural Network architecture performed well. In addition, prior

segmentation of mortar in the image is shown to have no impact on the classifier performance. Use of Class Activation Mapping shows that in most cases the classifier has learnt to ignore the mortar joint interface. The developed classifier performed equal to or better than current manual inspection procedures.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor, Dr Matt DeJong, for his encouragement, support, and guidance. I am thankful for the opportunity he provided for me to learn, grow, and develop throughout my PhD. I acknowledge both Trimble and the Engineering and Physical Sciences Research Council who provided for me financially.

I would also like to acknowledge Professor Cam Middleton, Dr Ioannis Brilakis, and Dr Rod Lynch for their advice and direction in the completion of this research.

I am thankful to my FIBE-CDT cohort 2 classmates. I value their companionship and friendship throughout our PhD pursuits. I am especially thankful to my friend, Mark Allen, who championed my wellbeing and helped me to stay positive and maintain progress through the difficult stages of my PhD.

I am incredibly grateful to my loving parents, John, and Gill Brackenbury, who supported my education and nurtured my interest in engineering from an early age. Additionally, I am appreciative of my uncle and Godfather, David Huddart, who has always inspired and fuelled my curiosity in engineering.

Most of all, I would like to thank my wife, Jessica Brackenbury, for her enduring support. Her patience, kindness and love made this possible.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

BCMI	=	Bridge Condition Marking Index
CAM	=	Class Activation Mapping
CANUPO	=	Caractérisation de Nuages de Points
CIRIA	=	Construction Industry Research and Information Association
CNN	=	Convolutional Neural Network
CWT	=	Continuous Wavelet Transform
DIC	=	Digital Image Correlation
HBIM	=	Historic Building Information Model
MVS	=	Multi-View Stereo
PSPNet	=	Pyramid Scene Parsing Network
RANSAC	=	Random Sample Consensus
R-CNN	=	Region Based Convolutional Neural Networks
RMSProp	=	Root Mean Squared Propagation
ROC	=	Receiver Operator Characteristic
SHM	=	Structural Health Monitoring
SFM	=	Structure from Motion
SGD	=	Stochastic Gradient Descent
UAV	=	Unmanned Aerial Vehicle
2D	=	Two dimensional
3D	=	Three dimensional
mm	=	Millimetre
Px	=	Pixels
f_t	=	Predicted probability of image window t being defective
o_t	=	Actual probability of image window t being defective
N	=	Number of image windows

LIST OF APPENDICES

1 INTRODUCTION

1.1 Background and need

Although construction of new masonry arch bridges is now not common, legacy masonry arch bridges underpin much of the world's transport networks. For example, 60% of the bridge stock of the major European rail administrations are masonry arch bridges or culverts which represents over 200,000 structures. In the UK alone there are 18,000 masonry rail bridges representing 47% of the rail bridge stock. Masonry arch bridges have proved to be durable, with the majority in Europe being between 100 and 150 years old. Furthermore, a significant proportion (approximately 12%) are over 150 years old (Orbán, 2004). Life cycle costs for masonry arch bridges are found to be more economical than for most other structure types. The vast majority of these 200,000 structures are in a good or medium condition, but there are a sizeable proportion (approximately 15%) which are in a poor or very poor condition. There is also a tendency for accelerated deterioration (Network Rail, 2015).

Effective management of civil assets is essential due to their age and susceptibility to rapid deterioration in adverse weather. Funding constraints result in the deferment of much of the planned renewal works on civil assets, making inspection and examination even more important. However, signs of imminent failure are often hard or impossible to detect by visual inspection, and it has been found that Network Rail, the rail infrastructure owner and operator in the UK, have been unable to deliver some examinations in accordance with the required standards (Davies and Dennis, 2021). Incidents such as the collapse of the Grove Nook Lane masonry arch bridge at Barrow upon Soar in England onto open railway lines of the Midland Mainline in August 2016 demonstrate the potential

for catastrophic consequences. In this case it was only human intervention, the last line of defence, which prevented a serious outcome. This also demonstrates the vulnerability of the current risk control and mitigation processes of civil assets, which are shown to be heavily dependent on the knowledge, competence and expertise of individual staff, increasing the vulnerability of those controls (Office of Rail and Road, 2017).

Network Rail aims to conduct a detailed condition survey of each bridge every six years. Historically, Network Rail have been falling short of this target. In the six-year period between 2001 and 2007, just 60% of the bridge population was assessed. This backlog was partially attributed to access problems caused by busy lines (Office of Rail and Road, 2007).

Therefore, there is a need to improve on the existing status quo. An automated visual inspection process can help increase the speed and frequency of inspections, as well as reducing the danger and disruption, and the subjectivity associated with current inspections. Such automated inspection can also feed towards a more impactful adoption of a Structural Health Monitoring (SHM) framework for bridges, where it is identified that the current lack of data is one of the key barriers towards implementation (Campbell et al., 2020).

1.2 Automated masonry inspection

Development of vehicle mounted imaging and laser scanning sensors have made the collection of visual and geometric data of bridge structures much easier. However, bridge inspection would still be subjective if the interpretation of such a dataset is performed by a human inspector (McRobbie, 2015).

Much progress has been made towards automating the detection of defects in concrete and asphalt infrastructure using both artificial intelligence and more hand-crafted methods. However, relatively little research has been focused on masonry. These concrete and asphalt studies have often shown satisfactory performance in conditions with few distractors, but performance was shown to deteriorate where conditions worsen. For instance, one study that used a deep learning neural network to distinguish defective from non-defective surfaces found that their classifier confused joints in concrete pavements for cracking (Gopalakrishnan et al., 2017).

Masonry suffers from many of the same defects as concrete, for example both feature spalling and cracking. Therefore, there is potential to apply similar automated methods,

both machine learning and hand-crafted, to masonry surfaces. A few studies have applied such approaches successfully, though mostly to clean masonry as seen on housing, where defects are much more discernible than in the imperfect masonry surfaces often seen on masonry arch bridges. Studies on masonry have also suggested that classifiers confuse mortar joints with defects (Chaiyasarn, Sharma, et al., 2018).

1.3 Research motivation and objectives

The previous sections detail the shortcomings of the existing manual practices of inspection of masonry arch bridges, and the existing progress that has been made towards automated inspection. This context provides the basis for the motivation of this work:

- Machine learning techniques hold promise for automated, image-based inspections of masonry arch bridges. However, we need a better understanding of the effect of mortar joints and other imperfections on the automated, image-based classification of defects in masonry, to improve the detection performance for the deteriorated surfaces typical of historic masonry arch bridges.
- An understanding of the performance of the existing manual inspection process for recording defects in masonry arch bridges is needed: for comparison to the performance of a state-of-the-art image-based automated classifier.

Based on these factors, the objective of this dissertation is:

> *To evaluate and quantify the ability of different defect classification methodologies, to increase the understanding of the factors key to the performance of automated defect classification on masonry arch bridges. This therefore enables the development of a state-of-the-art automated image-based classifier with performance exceeding that of the existing manual inspection process.*

To meet this objective, a variety of approaches will be examined as outlined in Chapter 1.4. These approaches are tailored to the different stages in the bridge inspection pipeline, from the creation of image textures through to the identification of the root cause of identified defects.

1.4 Outline of thesis

The approach taken to achieve the objective in Chapter 1.3 is as follows. This chapter introduces the problem and states the objectives of the thesis. Chapter 2 then reviews the

previous research in the problem area, understanding the defects present on masonry arch bridges, the current inspection methods including their pitfalls, and the existing data capture and automated defect detection methodologies.

Chapter 3 examines deeper the defects present on masonry arch bridges to determine the defect classes an automated classifier should focus on. A medical analogy is adopted to determine the implication of the detected defects on the health of the bridge. Visible defects are associated both with their underlying root cause as well as interventions required both to treat the bridge of its underlying deficiency and fix the visible defects. This analysis has been built into a connected map, such that future automation is possible.

Much of the work of this thesis investigates applying machine learning techniques to images. There is therefore a need for substantial amounts of annotated image data of masonry surfaces for both training and testing. The dataset that has been built for this is outlined in Chapter 4. Details of the case-study bridges are presented, along with the examination of different methods of creating orthorectified image textures of bridge surfaces. Focus is particularly applied to the curved surface of the arch barrel, where appropriate unwrapping of the surface is important to avoid masonry scaling in images.

Chapter 5 compares a hardcoded deterministic method with deep learning probabilistic methods, for detecting mortar joints in masonry arch bridge surfaces. Mortar joint location can aid in determining the root cause of identified visual defects through following the mapping developed in Chapter 3.

Following the detection of mortar joints, the detection of defects is examined in Chapter 6. The focus is on the effect of mortar joints on the performance of a defect classifier. Several classification strategies are examined, incorporating mortar joint information in different ways with the resulting classification performance compared across different datasets.

Chapter 7 applies the findings of Chapter 6 to test the defect detection performance of a state-of-the-art classifier against that of the current manual inspection process. This is compared across a dataset of three different bridges whereby the defects identified in a detailed inspection are compared to the ground truth and those identified by the automated classifier.

Finally, Chapter 8 summarises the work completed and the conclusions of the previous chapters. These results provide suggestions for potential future research opportunities for

both the improvement of the automated defect classifier, and for further automation of the inspection pipeline.

2 LITERATURE REVIEW

2.1 Masonry arch bridges

Figure 2-1 shows the typical terminology used for the main structural elements of masonry arch bridges. The arch of the bridge spans the space between the two abutments, with the backfill material supporting the bridge road surface being contained by the spandrel walls and wing-walls. Briefly the main elements are (Scotland's Oldest Bridges, 2021):

- Abutment: a solid masonry structure built to support the lateral pressure at both ends of an arch.
- Arch Barrel: the whole of the masonry arch ring. The lines where the side edges of the barrel meet the face are called the intrados (bottom edge) and extrados (top edge). The depth, or thickness, of the arch is the distance between them. The lower surface of the barrel is called the soffit and the upper surface the arch back.
- Back fill: the filling material in the space above the arch barrel between the spandrels. It provides a firm, flat surface for the top decking of the bridge. Often the back fill consists of compacted material excavated during the building of the bridge foundations. Water in the backfill freezing leads to expansion which can push out the spandrels. Therefore, weep holes are used for drainage, and the surface is often waterproofed.
- Crown: the centre, or the highest part of the arch barrel.
- Haunch: the part of the arch barrel above the springing line but below the crown.
- Parapet: the wall on either side of the bridge deck, joining the spandrel walls at the brick deck.

- Pier: a masonry structure to support the vertical load of adjacent ends of two arch spans on a multi span arch bridge.
- Rise: the height of the arch barrel, the difference in height between the springing line and the crown of the arch.
- Span: the length of the arch barrel, the horizontal distance between the springers each side of the arch.
- Spandrels: the parts of the bridge directly above the haunches but below the crown level. An open spandrel will have gaps rather than walls in this area. The spandrel walls stiffen the barrel of the arch, but only at its edges (where the walls are) which could lead to the centre of the barrel being more flexible than its edges.
- Springer: the lowest brick course of the arch barrel sitting on the foundation or abutment. The springers run across both sides of the barrel at the lowest level of the soffits. The springing line is the line at the lowest edge of the springers.
- Wing-wall: a retaining wall splaying out from the abutment retaining the fill of the approach to the bridge.

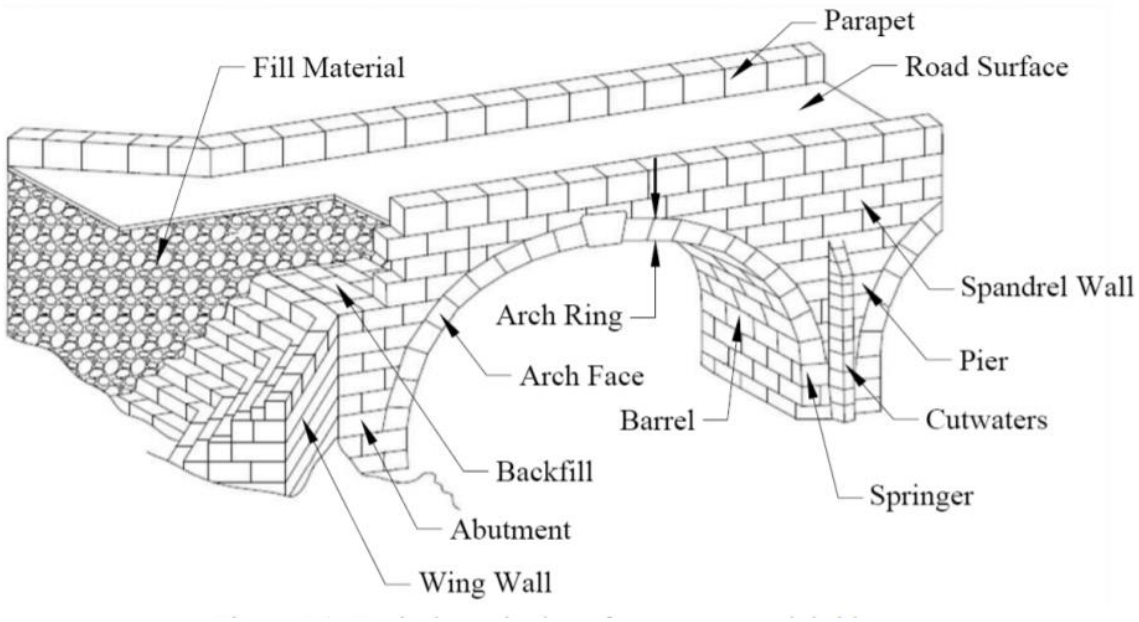


Figure 2-1: Typical terminology for masonry arch bridges (Siwowski, 2015)

2.1.1 Masonry arch bridge defects

Typically, the problems faced by arch bridges are a result of deterioration in the condition of the foundation, the structure, or the materials of the bridge. This deterioration manifests itself in the form of (Pitchford et al., 2003):

- Distortion of the arch structure: This can lead to transverse cracking.
- Spandrel wall separation: This is where the spandrel wall (the wall built on the curve of the arch filling in the space between the arch and the bridge deck) moves away from the arch barrel.
- Ring separation: This is where the layers of bricks in the arch ring delaminate from one another.
- Backfill failure: This could be a result of the backfill becoming saturated.
- Water damage: This causes freeze-thaw or leaching damage leading to deterioration of bricks and mortar.
- Chemical damage: This is mainly caused by attack from various sulphates.
- Physical erosion
- Vegetation: Growing in the brickwork, displacing, and breaking apart the masonry structure.

Therefore, the main visual indicators of bridge problems are, deterioration of the condition of the bricks and the mortar, fractures and cracking in the brickwork, salt efflorescence, and vegetation.

Cracking in masonry is caused by movement that cannot be prevented but can be accommodated. In flexure, cracks in masonry bricks first occur at about 80% of failure strength. Cracks in mortar could be a result of differential movement between brick and mortar, for instance due to different coefficients of thermal expansion. Cracks can also be induced between walls above and below grade, for instance between the parapet and spandrel walls, as walls above grade are more prone to volume change caused by heat, moisture or freezing. Foundation movements can be caused by changes of soil water content with the seasons or due to vegetation. Uneven foundation settlement most often results in diagonal or vertical cracks in masonry. Relatively small vibrations, such as vehicle movements over bridges, can lead to built up stress concentrations in masonry, resulting in unexpected cracking. Some cracks in masonry are also present at the time of

construction. For instance, mortar shrinkage upon drying can result in cracks, and small cracks on brick faces could be caused by inadequate control in brick manufacturing which may penetrate only slightly or could extend through to the core of the brick. Cracks in masonry are a source of masonry performance failure and water permeance, though tests conducted at the Building Research Establishment in England have shown that stepped cracks up to 25mm wide only reduce the capacity of brick walls to carry vertical loads by less than 30% (Grimm, 1988).

Laminations are cracks parallel to the face of the brick, so are not visible on the surface. All extruded bricks are laminated to some extent. Although there is no evidence that laminations affect brick performance, weathering and compressive stresses may cause the brick face to spall at these laminations (Grimm, 1988). The initial damage from spalling is characterised by a few cracks, leading to severe cracking, flaking, and spalling which may gradually extend deeper into the material causing strength loss and ultimately total disintegration of the brick (Larbi, 2004).

Efflorescence is caused by the acid soluble constituents of bricks dissolving and being transported to other sites where they recrystallise to form new compounds. Where recrystallisation is on the masonry surface, this is termed efflorescence, and appears as a white coating. Efflorescence is usually just an aesthetic form of deterioration, though can eventually lead to surface deterioration of the masonry itself as a result of material loss caused by spalling (Larbi, 2004). The presence of water therefore plays an important role in the deterioration process of stones (Saiz-Jimenez and Marszałek, 2004).

2.1.2 Masonry arch bridge visual condition assessment process in practice

The specifics of bridge inspection programmes are different in different administrative regions, though most follow similar procedures and schedules. Generally, there are four main types of bridge inspection that are performed as part of a bridge inspection programme. These are, routine surveillance, general (overview) inspections, principal (detailed) inspections and non-routine inspections. Staff continuously perform routine surveillance during their normal duties when they pass through the bridge site. It involves a brief check for obvious defects in the structure. General inspections are performed at intervals of between one and three-years. These involve a visual inspection, checking for any significant signs of deterioration since the previous inspection. The bridge is observed from ground level and the results are documented. Principal inspections are performed at intervals of between five and six-years. They involve a detailed close visual examination

from within touching distance of the structure. This is supplemented by other investigations and measurements as appropriate for the structure (e.g., hammer tapping). The principal inspection is used to inform maintenance planning. Non-routine inspections are conducted in case of doubt about the actual condition or capacity of a bridge. This could be a consequence of something discovered during a routine inspection, or because of something happening to the bridge (e.g., a bridge strike). These involve a visual inspection as well as other required specialist investigations of the structure (Atkins, 2009; Helmerich et al., 2007).

Visual inspections are used as the first level of inspection. The results determine if additional more detailed inspection or maintenance is required. As part of the visual inspection, it is recommended that photography is used to record the general appearance of the structure and any defects or evidence of deterioration. Photographs are thought to be effective for communicating the bridge condition information to others for comparison of features and showing the deterioration of the bridge over time. They also decrease the subjectivity of the inspection process. Inspectors carry photographs and documents of the findings from previous inspections with them and use these to identify whether existing defects have changed and identify any new defects. For photographs to be effective for comparison of a feature, they should be taken with a measure of scale, and the position and angle of the view should be consistent between different inspections (McKibbins et al., 2006).

The following outlines information that is recorded during a visual inspection undertaken in the UK, which is similar to that recorded elsewhere (McKibbins et al., 2006):

- All the basic dimensions of the structure
- The material type and the form of construction
- Any evidence of an impact
- A map of the defects found in the structure
- Any cracks in the structure. The position, orientation in 3D space, length, displacement (crack width), and whether the crack has historically been repointed are all recorded. The displacement of the crack may not be constant across its length, so this is checked. Additionally, cracks may only open when the bridge is loaded, so inspectors look for evidence of closed cracks since most inspections are conducted when the bridge is not loaded. Cracks in a road

surface for road bridges are also recorded as these can give an indication of movements in the backfill.

- Historical repairs made to the bridge. This is because these may hide ongoing defects. For example, cracks could have been hidden by repointing making them impossible to detect. An extra thick mortar joint may provide evidence that this is the case, particularly if the mortar is obviously different to surrounding mortar.
- Any other evidence of movement of the structure. This can be the distortion of regular shape, misalignment, tilting, bulging, hollowness, or excessive movement under load. Such movements are measured using simple manual techniques, for example a line and a plumb bob. This is important to record as often masonry can be subjected to quite large differential movements without showing evidence of cracking.
- The degree of any masonry wetness. This is objectively recorded on a scale from dry to running water as this gives an indication of the level of the water table in the backfill. Weather conditions, both at the time of inspection and in the prior period are logged as these can influence the presence of water on the bridge.
- Any evidence of the effect of water on the fabric of the bridge. This is especially the case for bridges over water which are checked for visible evidence of scour.
- Mortar loss and the condition of the mortar. This gives an indication of the strength of bond between the mortar and the masonry, and the ability of the masonry to transfer load. This is assessed by the depth of any open joints, their location, and the approximate proportion of the structure with open joints. If masonry has been repointed since the last inspection, the depth of the repointing is determined as well as the condition of the mortar behind the repointing. Additionally, the presence of any voids between the repointing and original mortar are noted.
- The presence and extent of any obvious signs of leaching. This is a whiteish mineral deposit on the masonry surface. Any other evidence of a chemical attack is also recorded.
- The condition of the masonry fabric. This includes noting the extent and depth of any spalling, erosion, softening or deterioration of the masonry. Additionally,

the presence of any vegetation and its extent is recorded. Colonies of micro-organisms such as lichen can indicate internal deterioration of the masonry.

Whilst collecting field data from a viaduct carrying the East Coast mainline in Peterborough, the chalk markings shown in Figure 2-2 were observed, highlighting cracks identified in the masonry. This shows an example of the existing manual process an inspector conducts to identify and document the defects in the brick structure as part of a principal inspection.

As part of the investigations of this PhD, a general inspection of Harringworth Viaduct in Northamptonshire, UK was observed. This is much less detailed than the principal inspection detailed above in Peterborough. Here the inspector walked the length of the viaduct, both at ground level and track level, observing with binoculars and photographs the progression of defects that have been previously reported. A general inspection is therefore unlikely to detect any new defects on the structure.

Figure 2-2: Image taken whilst collecting field data showing marking involved in manual visual inspection

2.1.3 Inspection Subjectivity

Visual inspections are ideal for gathering information of the condition of a large bridge stock as they are simple, rapid and can be undertaken with minimal disruption. However, the main weakness is the reliance on visible features and subjectivity of observations. Early signs of structural distress may be easily overlooked or perceived as inconsequential, particularly when larger potentially longstanding defects are present. Whether such defects are recorded therefore depends on the skill and knowledge of the individual inspector, reducing confidence in inspection results (McKibbins et al., 2006). In the UK, concern has been expressed over the management of the bridge condition monitoring process, particularly highlighting the accuracy of the process (Office of Rail and Road, 2007).

The Federal Highways Administration of the USA initiated a major study to examine the accuracy and reliability of documentation generated during routine visual inspection of highways bridges (Phares et al., 2004), acknowledging that the inspections heavily rely on subjective assessments by bridge inspectors. In this study, 7 routine inspection tasks akin to a general inspection as described in Chapter 2.1.2, were performed on 6 bridges by 49 Department of Transport bridge inspectors from 25 states. The bridges were of both concrete and steel construction. The bridge inspectors assigned condition ratings for each bridge, with associated notes and photographs as they would normally do. Additionally, a ground truth reference rating was assigned for each bridge based on the defects observed by those conducting the study.

In terms of accuracy, the average condition rating assigned to 13 of the 18 primary components (deck, superstructure, substructure) across the 6 bridges were better than the reference, with the remaining worse. For 14 of the 18 primary structural components examined (78%), the average assigned condition rating was statistically different from the reference at a 5% significance level. For individual inspectors, if the reference ratings are correct, 58% of their assigned ratings are assigned incorrectly. Condition ratings are assigned on a 10-point scale between a failed/condemned and an excellent/as new condition. Compared to the reference rating, the average inspector condition rating is 0.55 points better for the decks, 0.24 points better for the superstructures, and 0.08 points worse for the substructures, making the overall average 0.24 points better. This suggests a tendency to overestimate the condition of certain parts of the bridge.

In terms of reliability, it was found that on the 10-point condition rating scale, on average between 4 and 5 different ratings, with a maximum of 6 and minimum of 3, were assigned

to each structural component. Across the different structural components, 32% of condition ratings were found to be more than 1 point different from the average assigned rating, and 5% more than 2 points different. While variability was most prominent in terms of condition ratings, it was also present in other inspection documentation such as photographs and field notes.

It was identified that 13 photos would be required as the minimum number to fully document each bridge. On average inspectors took 7 photos (standard deviation 3.8, max 19, min 1) which included instances of multiple photos of the same thing. The authors identify difficulty of access as a reason some inspectors skipped some of the photos. On average across the different inspectors, only 31% of the features expected were photographed. It was found that while most inspectors use photographs to document the overall condition of a structure, they are used less frequently to document specific structural deficiencies. Across the 7 bridges there were 20 defects that should be recorded, but on average only 69.4% of these were noted. Therefore, it was suggested that many inspectors may not be completely documenting important modes of deterioration in their notes. In this study, inspections were conducted under direct observation, and inspectors knew that they were under test conditions knowing their inspection results would be closely examined. Additionally, the inspectors were highly trained and experienced professionals. The inspections are therefore likely to be more detailed than normally would be the case, yet there was still significant variability in the structural condition documentation reported.

In conjunction with this study, the accuracy and reliability of in-depth inspections was also examined in the context of correctly noting the presence of known defects (Graybeal et al., 2001). In-depth inspections, akin to principal inspections described in 2.1.2, are close-up, hands-on inspections of the structures, so ladders and man lifts were provided.

Firstly, steel superstructure bridges were examined and 42 of the previous 49 inspectors performed this study. Generally, it was found that most inspectors were reporting the recurring defects, such as paint system failure and corrosion, which are noticeable throughout the structure, but a much lower proportion of inspectors noted local defects. For instance, only half noticed bearing misalignment, 3 a crack indication on a weld, 2 a missing rivet head, and 7 impact damage. On one bridge, the overall accuracy rate for correctly identifying crack indications was only 3.9%. Inspectors were monitored while carrying out their inspections, and the thoroughness with which they examined the welded connection details was recorded. It was found that only 45% of the inspectors examined

the welded components with enough thoroughness to be considered to have completed a thorough in-depth inspection – these inspectors made up 86% of those who correctly identified a crack. 36% of the inspectors were considered to have performed an incomplete in-depth inspection due to their lack of thoroughness examining welded connections – none of these inspectors correctly identified any cracks. It was found that those inspectors who belonged to the first group, who thoroughly inspected welds, tended to take longer over the inspection and be more mentally focused. During everyday inspections, not those undertaken as part of a study, inspectors are unobserved and potentially under greater time pressures. They are therefore more likely to take less time, being less thorough and as a result more likely to miss potential defects.

Secondly, delamination assessment was performed on a concrete bridge deck using mechanical sounding and other visual inspection techniques by 22 teams of 2 inspectors. 2 of the teams only provided a numerical estimate of the delaminated deck area, whereas the other 20 also provided maps showing locations of delaminated areas. Only 5 team's predictions were within 5% of, and 14 team's within 10% of the ground truth delamination proportion of the bridge deck determined by the study authors. For 3 teams, it was obvious that they failed to detect large areas of the delaminated deck. Additionally, it was found that the individual delamination locations were inaccurately positioned on the delamination maps. 69% of the deck area was indicated as delaminated by at least one team, and only 1% indicated as delaminated by at least 15 teams, with no areas found to be delaminated by all the teams, showing little agreement between the teams. It was therefore suggested that very few teams provided results that portrayed the condition of the bridge accurately. The authors therefore concluded that when an in-depth (or principal) inspection is prescribed, such as in the previous two examples, it may not give any findings beyond those that could be found with a routine (or general) inspection. This is despite in-depth inspections often being prescribed primarily to detect those types of defects, such as cracks in welds, which this study has shown are rarely detected.

A further study looking at the visual inspection of highway bridges was performed for the Department for Transport in the UK (Wallbank and Department of Transport, 1989). Here the categorisation of the condition of 200 bridges based on visual inspection data was compared to chloride levels in the bridges, an indication of potential corrosion in steel reinforcement. It was found that some of the most worrying defects could cause severe damage to the bridge before any visible signs were apparent. In one bridge the visual inspection classed the bridge as of good condition, whereas test data suggested it probably

to be in poor condition. In this study it was noted that, “when recording and comparing the visual condition of a wide variety of bridges it is difficult to be precise and consistent.”

Most recently, Bennetts et al. (2018) examined the recording and rating of defects recorded across 200 highways bridges in the UK. The results from the scheduled routine principal inspection for each bridge (carried out between 2014 and 2016) were compared to a second independent inspection where the inspector identified and rated defects but did not carry out a full inspection. Considerable variation was found in the recordings of defects between the two inspectors, in that almost 30% of the defects recorded by the second inspection could not be identified with an equivalent in the principal inspection report. With those defects that both inspections recorded, there was only a 34% agreement in terms of the severity and extent of the defect. This study therefore highlights both the variability in the detection of defects, and the classification and grading of defects in the visual inspection process. This variation is despite preceding efforts to minimise the subjectivity of the visual inspection process through ensuring wherever possible observations are quantitative or assigned to an indexed value, and that the inspection follows a clearly prescribed comprehensive range of observations (McKibbins et al., 2006).

Factors that affected individual bridge inspection results were examined as part of the Federal Highways Administration study (Moore et al., 2001). A consistent relationship was found between inspection results and the inspectors near visual acuity, their fear of passing traffic, and their formal training, as well as the proximity from which the inspector examined the structure, and their perception of the bridges maintenance, accessibility, and complexity.

Further examination of factors that may influence inspector accuracy was conducted by Megaw (1979) in order to design tasks to improve inspection performance. The focus was on inspection tasks in a laboratory environment, which to a degree simulate real bridge inspections. Factors have been categorised into subject factors, physical and environmental factors, task factors and organisational factors. Subject factors are based on characteristics of the inspector and include visual acuity, scanning strategies employed, age, personality, sex, and intelligence. Physical and environmental factors include, lighting background noise, music while working, workplace design, and workplace aids (such as magnification). Task factors include time for inspection, viewing area, shape of viewing area, density of items, spatial distribution of items, fault probability, fault mix, fault discernibility, and product complexity. Organisational factors

include, number of inspectors, instructions, feedback/feedforward, training, standards, time-on-task, rest pauses, shift, sleep deprivation, motivation, incentives, job rotation and social factors such as isolation of inspectors. It is identified that there is a large interaction between the different factors, for instance optimal luminance levels depends on the age of the inspector. While some of these factors have already been examined in relation to bridge inspection above by Moore et al. (2001), all have the potential to affect the outcome of an inspection, and many are hard to control in the field.

One key factor with potential to influence inspection outcome is the number of potential defects. As the number of potential defects increases, the inspection reliability is found to decrease. Additionally, features in the inspection sample that are not actually defects but do have a different appearance from the general inspection sample, can cause inspectors difficulty as they require a decision as to whether to record it. This is more challenging for an inspector than encountering a defect where there is no doubt about whether to record it (Sheehan and Drury, 1971). With bridge inspections, particularly with masonry bridges, there are many such scenarios where non-defect and defect features can have a similar appearance.

McRobbie (2015) conducted a study looking at the ability of people to visually detect features in charts from set distances in comparison to within digital images of set resolutions, to determine whether bridge inspections could be more effective if performed on digitally imaged structures as opposed to in person. 40 participants undertook the study, where they viewed charts depicting, varying letter sizes, varying background contrast and varying line thicknesses and orientations, from distances of 3m, 6m, and 12m, and in digital images of 2 pixels per mm and 1 pixel per mm resolution. An obvious conclusion was that at both the closer distances and the higher resolution digital images, identification performance improved, yet the author identifies that when observing bridges in person it is often difficult to get close. For instance, when there are obstructions 12m may be as close as an inspector can get, when inspecting parts of the structure such as the soffit from a footpath 6m is a representative inspection distance, and even when access is good a 3m inspection distance will be typical due to the height of the bridge. It was also found that participants had little difficulty identifying cells in the charts with no lines even at a distance, suggesting that false negative detections are much more likely than false positive, meaning inspectors are much more likely to miss a defect than incorrectly find a defect. Comparing the digital and in person approaches to inspection, it is concluded that more detail can be seen in digital images at 1 pixel per mm than in

person at a separation distance of 3m. Additionally, lines as thin as 0.07mm wide were correctly identified at both 1 and 2 pixels per mm resolution, whereas at a 12m separation distance, only 44% of 0.42mm thick lines were correctly identified. These results demonstrate that when performing inspections in person, inspectors are likely to fail to identify important features in the bridge, and there is a greater chance of detecting those features in images of bridges.

Laefer et al. (2010) has also compared digital and in person approaches to manual inspection. They have compared the accuracy of laser scanning and photographic based digital approaches with ground based and elevated in-person inspection for detecting cracks in building facades. Ground based inspection is akin to the process of general inspections of bridges, and elevated inspection akin to principal inspections. The study was undertaken by 2 different inspectors on 4 buildings between 8.2 and 14.3m high, of both brick and concrete construction. It was found that manually analysing digital photography was most accurate, detecting 46% of the total cracks, but was found to underestimate the length of cracks by 50%, i.e., cracks which were detected were not detected in full. In comparison, ground based and elevated in-person inspection detected 13% and 31% of cracks, respectively. The laser scanning digital approach was found not to be particularly accurate particularly higher in the building due to the scan points spreading as distance from the sensor increases. Digital photography inspection was found to be the most consistent methodology across the height of the buildings, whereas the accuracy of ground based in-person inspection greatly decreased above 2m.

From a consultation of bridge inspectors, McRobbie (2015) has identified that no inspectors routinely collect a full image set covering all visible parts of the bridge being inspected. Therefore, for unimaged areas of bridges with no defects recorded, there is no permanent record of the bridge generated. This means if a defect is subsequently detected in a future inspection, there is no way of telling if the defect is new, or longstanding and was previously overlooked.

These different studies highlight the main deficiencies with the current bridge inspection process from a Structural Health Monitoring (SHM) perspective, namely the discrepancy between the reporting's of individual inspectors as to a bridge's condition and the lack of a permanent record of the structure. Technological advancement has come some way to addressing this second deficiencies as will now be discussed.

2.2 Data capture methodologies

There has been great technological advancement in the development of data capture methodologies for inspection of existing infrastructure. In this chapter, some of the current state-of-the-art equipment and methodologies are examined together with how they can be applied to masonry arch bridges.

While conducting work for this thesis, an example data capture system has been observed being utilised by the East-West Rail alliance scheme. This is a major infrastructure project in the UK aiming to re-establish a railway connection between the cities of Oxford and Cambridge (East West Rail, 2021). To capture the condition of existing assets on the route to aid with planning the project, a Leica Pegasus Two scanner was used. This is a vehicle mounted scanner which is driven around the study area. The unit includes a laser scanner and an option of either six or eight digital cameras to capture the full 360-degree view and generate visual 3D models of the assets being scanned. To track the scanner's location, it uses both GPS and inertia sensors (Leica Geosystems, 2014). Figure 2-3 shows an example of the visual data captured when the scanner was driven under a masonry arch bridge on the route. In this case the scanner was driven at a speed of between 30 and 40mph. Figure 2-3b shows that it is possible to identify mortar joints from the data captured at this speed with this scanner, but the accuracy of defect detection would probably be limited by the blurry and low-resolution output.

An experimental data capture system for tunnels was developed as part of the DIFCAM (Digital Imaging for Condition Asset Monitoring) project, a collaboration between Omnicom Engineering, Atkins and the UK's National Physics Laboratory (Parker, 2013). This was intended as a demonstrator to show the feasibility of imaging and laser scanning based techniques for tunnel examination, to capture a complete visual record as well as a 3D model of the tunnel. The system relies on a road rail vehicle platform which carries an array of 11 24mp DSLR cameras with 4 high powered flash units covering the full profile of the tunnel for image data, as well as a laser scanner for geometry data, and GPS, inertia, and tachometer sensors for positioning data. This can work at speeds of 1m/s (2.2mph), capturing imagery generally better than 1mm per pixel resolution and geometry shape resolution of 10mm with accuracy of 1mm. Their intention is that technological development will make it possible to collect data at line speed in the future so the system can be integrated into the Network Rail New Measurement Train.

a)

b)

Figure 2-3: Leica Pegasus 2 visual data output at 30-40mph showing: a) a 360-degree image view, b) a full resolution view of the masonry on one of the bridge abutments

A different commercial data capture system for capturing visual 3D models of tunnels has been developed by Pavemetrics called the Laser Tunnel Scanning System (Laurent et al., 2014). This can capture 1mm per pixel resolution images and 3D data at scanning speeds up to 20kph (12mph), so it is therefore able to visualise and measure features with sub mm accuracy. This uses multiple high speed laser profilers which work using the principle of triangulation, where a high intensity laser line is projected onto the structure and recorded by a digital camera. The height and reflected intensity of each pixel are then recorded to form continuous images of the surface.

A more recently developed commercial data capture system is called the Dibit high speed scanning system, also developed for capturing visual 3D models of the condition of tunnels (Mett and Eder, 2019). This produces a true colour high resolution 3D model of the tunnel surface using an array of up to 10 high speed industrial cameras, each capturing up to 30fps, alongside a high-performance LED lighting array. The use of high-speed cameras in combination with the bright LED lighting array mean that it is possible to capture shake free photos with 1mm per pixel resolution in dark tunnels at speeds of up to 50mph where cracks as small as 0.3mm in thickness are visible. The Dibit system is modular, so additional sensors can be added as well as the camera array, for example laser scanners or thermal and multispectral cameras. 3D geometry is therefore generated through either photogrammetry as standard, or a hybrid system where a laser scanner is included in the system.

Whilst the above vehicle mounted visual and geometrical data capture systems have proven to be effective at generating visual 3D models of tunnels, for bridges their utility is somewhat limited. For all four of the systems, where surfaces of the bridge are perpendicular to the road, such as wing and spandrel walls, the separation between the imaging sensor and parts of the surface can be greater. This leads to lower image resolutions, and greater camera focusing problems for these parts of the surface than for the parallel parts of bridges such as the soffits, which are similar in geometry to the tunnel linings the sensors were designed for. Additionally, the quoted acquisition speeds may not be practical for imaging bridges due to the need to focus imaging sensors onto the bridge. Whereas the camera to tunnel lining distance remains relatively constant, the separation distance will be different at a bridge compared to the surrounding environment. For instance, the DIFCAM project acknowledges it takes them roughly 30 minutes to align the sensors on the vehicle before entering a tunnel, though suggests this could be reduced with automation.

As a result of these limitations, alternative data capture solutions have been explored for bridges. McRobbie (2015), has proposed an image capture system for bridges using a DSLR camera with in-body image stabilization and an automated pan-tilt unit for creating stitched image panorama photos of bridges. This system is moved around the bridge to capture the bridge from different viewpoints, using different length lenses for different camera positions to ensure the bridge surface has been captured with sufficient resolution. All images taken are re-projected as orthophotos of the bridge (photos taken such that there is no perspective distortion), and then neighbouring photos aligned to create stitched panorama orthophotos of the bridge components. The system incorporates a distance measurement laser unit for measuring the perpendicular distance to the bridge from each imaging position to aid in image re-projection, as relying on image features for alignment has potential for error due to few or non-unique features in bridges. However, currently this system is quite slow to collect data – for a simple bridge with two wing-walls, 12 imaging positions would be needed, which would take about 1 hour for imaging as well as 4 hours of setting up and measuring positions. Admittedly, this time could well be reduced in a commercial version of such a system, though it is considerably longer than it would take to conduct a general inspection of the bridge in person. Additionally, extending the orthophoto stitching panorama approach to masonry arch bridges has proven to be problematic, as the arch of the bridge makes the re-projection of the images more complex given the image normal will be different for each point along the arch.

An alternative method of data capture used commercially on bridges is using Unmanned Aerial Vehicles (UAVs). These have been used to collect image and geometry data of masonry structures and bridges in the literature by Ali et al. (2019), Chaiyasarn, Khan, et al. (2018), Chen et al. (2019), Ellenberg et al. (2014), and Hallermann and Morgenthal (2016).

Chen et al. (2019) have used photographic data captured from a UAV to build photogrammetric 3D models of bridges. They identified a framework for evaluating the accuracy of the generated 3D bridge model based on parameters like occlusions or lack of image overlap causing incomplete data, outlier noise caused by texture less backgrounds like the sky, non-uniform point density, surface deviation, and geometric deformation.

Ellenberg et al. (2014) investigated the use of UAVs for crack detection in masonry. In images captured with a UAV mounted camera flown 90cm from a loaded masonry wall in a lab test, they were able to see the formation of cracks. However, they identified that

flying close to structures means the GPS readings of UAVs are more likely to be inaccurate and the UAVs are more likely to feel the effects of wind, where a gust could cause the drone to collide with the structure. They found that with a 16MP camera positioned 12m from the surface, only lines greater than 19mm thick could be observed suggesting only a low-resolution image is achieved.

Hallermann and Morgenthal (2016) performed a UAV based survey to manually inspect a masonry arch viaduct, recording surface images with a mean resolution of 1.1mm per pixel, which is much better than that achieved by Ellenberg et al. (2014). They produced a geo-referenced, true scale, coloured photogrammetric 3D model of the bridge. Orthophotos of the 3D model were used to create architectural drawings and visually identifying damage in the bridge. Damage detected in orthophotos could then be identified in the 3D model to measure the extent of the damage.

The current methodology used both in the literature and commercially to generate 2D visualisations of 3D visual models of masonry arch bridges for the identification of damage, is to capture orthophotos of views of the model. This is appropriate for reinforced concrete bridges as the surfaces are flat, but for masonry arch bridges, the arch barrel is not flat, and therefore capturing the arch barrel with an orthophoto will lead to scaling in the 2D image, resulting in some parts of the barrel being depicted at a lower resolution than others. Figure 2-4 shows image data acquired from Network Rail of a bridge in their inventory. This depicts a 2D orthophoto of a bridge soffit taken from a 3D model of the bridge acquired using photogrammetry on UAV image data, suggesting this is currently the way 3D data of arched bridges is viewed commercially. Scaling can be seen in this image in that the brick courses to the left and right sides of the image are much narrower than those in the centre. In reality, the brick course thickness would be the same throughout the soffit. In the literature, orthophoto visualisation of arch barrels in 3D models is shown in the works of Hallermann and Morgenthal (2016) as already discussed and Riveiro et al. (2016). No work in the literature examined has identified a methodology for visualising the 3D geometry of masonry arches in 2D without resulting in scaling of the arch barrel.

So far in this chapter the focus has been on the collection of geometry and image data of masonry arch bridges. Orbán and Gutermann (2009) have investigated the use of alternative non-destructive testing data that can be acquired from masonry arch bridges. They have found infrared thermography able to provide useful, though only qualitative information on moisture and wetness in masonry as well as the degree of masonry surface

delamination. However, they acknowledge that this only gives information of up to a few centimetres below the surface. For identifying deeper, non-surface defects such as flaws in the masonry or ring separation, ground-penetrating radar, or sonic methods are required, however unlike all the methods previously examined above, these methods require contact with the masonry.

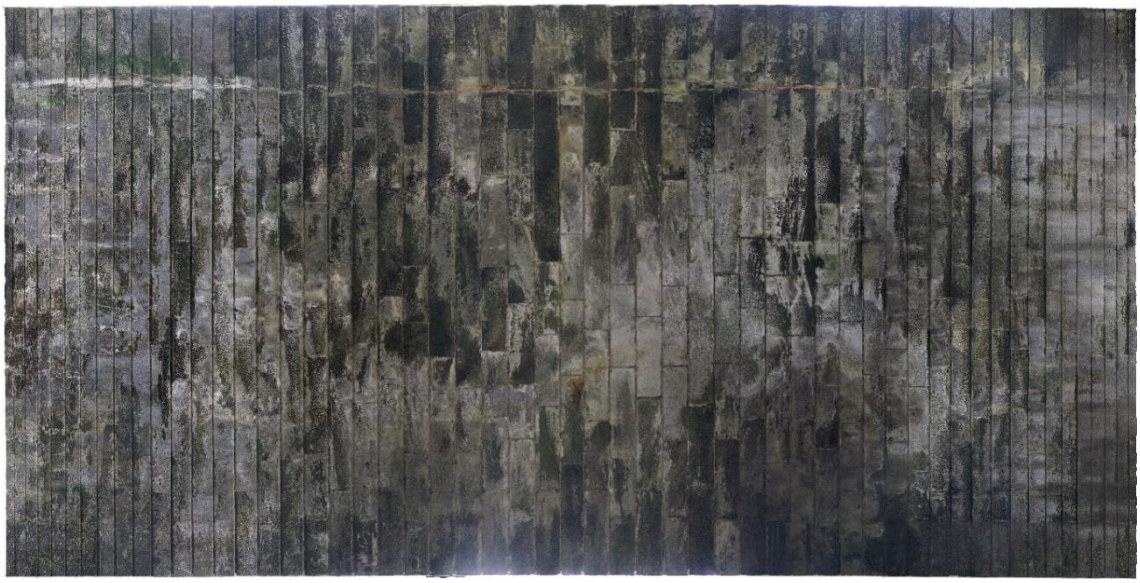


Figure 2-4: Soffit of arch barrel from Network Rail image data showing scaling towards the edges

2.3 Automated defect detection methodologies

The increased ease of collection of image and geometry data of bridges as examined above, coupled with the increase in performance and ease of access to processing power has led to much interest in automating the detection of defects from the data collected of infrastructure. McRobbie (2015) identifies that regardless of the quality and detail contained within image data taken of bridges, the inspection results will still be subjective if a human inspector performs the interpretation of the data.

Historically, much of the work to automatically identify defects in data has focused on reinforced concrete and asphalt-based infrastructure. These infrastructures generally have a higher degree of homogeneity in their surface textures when compared to masonry as they are only made of one material. In masonry, for comparison, the mortar adds an additional degree of complexity, where it is distinct both visually and in terms of geometry to the brick it surrounds. Some of the key developments in concrete and asphalt defect detection are first examined, followed by a comprehensive review of existing masonry defect detection research.

2.3.1 Defect detection in non-masonry surfaces

The focus of research to automatically detect defects in non-masonry infrastructure has shifted over time from approaches looking at hand-crafted models to exploit the contrasting features of defects, to machine learning and most recently to deep learning approaches. Much of the research focus has been placed upon the detection of cracks, and to a lesser extent, spalling and delamination.

Koch et al. (2015) examined many of the proposed approaches for assessing the visual condition of infrastructure, focusing on those approaches looking at asphalt road pavements and concrete in bridges, tunnels, and underground pipes. They identified that the approaches can be broken down into pre-processing methods (e.g., shadow removal), feature-based methods (e.g., edge detection), model-based methods (e.g., percolation, line tracing, neural networks), pattern-based methods (e.g., principal component analysis, support vector machines), and 3D reconstruction (e.g., structure from motion). The individual methods build upon each other, so a particular approach may employ several methods. For example, a feature-based method can be used to extract features of interest that can then be learned in a pattern-based method. The authors therefore identify that most approaches follow the steps of image acquisition, pre-processing, segmentation, feature extraction, object recognition, and finally, structural analysis. They have also identified the drawbacks of different approaches to defect detection. Thresholding based approaches face subjectivity in where to set a threshold, as basing a threshold value on prior knowledge limits the ability of the approach to be generalised. Pattern-based approaches rely on training data to set up a robust classifier, but as data is usually manually labelled, there is subjectivity in the labelling.

Early work looking to automate the detection of defects has focused on the feature-based methods identified in Koch et al.'s analysis above. That is to highlight features of the data that differentiate a defect from the surrounding background, exploiting known properties of the defects being sought.

One such feature of defects in image data is an edge. Abdel-Qader et al. (2003) compared the effectiveness of four different crack detection techniques based on highlighting edges in the intensity of images of concrete surfaces, those being the Fast Haar Transform, Fast Fourier Transform, Sobel, and Canny. By thresholding the intensity of the output, they determined whether a particular image contained a crack or not. This approach meant that some noise in the output was acceptable to still result in a correct classification. Testing on an evenly split sample of 50 images they found that, of the approaches tested, the Fast

Haar Transform was significantly the most reliable with 86% accuracy. Both the Haar and Fourier transforms are frequency-based approaches, converting the image from the spatial domain to the frequency domain, where edge features are higher frequency components, but the Fourier Transform performed the worst with 64% accuracy due to it picking up texture in the concrete. Canny and Sobel are both convolution filters used to highlight edges using the gradients of intensity changes across the image. Although both are susceptible to image noise, Canny looks to improve on Sobel through initially convolving with a gaussian mask to blur the image, thus reducing noise. Of the false detections, it was found that there were far more false positives than false negatives in the detection of cracks, suggesting that these techniques were all limited by their susceptibility to noise in the image.

Another approach looking to exploit known features of cracks takes advantage of their linear nature. For this Dijkstra's minimal path selection algorithm is used to search for cracks in pavements as a path composed of darker pixels (Amhaz et al., 2016). Potential endpoints are proposed for possible crack paths as the local minima of image intensity and minimal paths are computed between them without any direction or length constraints. The cost function is instead based upon the image intensity in the path. The cost function of a path within a crack will be lower than any other minimal path within the image background as the crack pixels are darker. At a global level, across the image, the locally proposed minimal paths are thresholded based on their cost function such that only those likely to lie in a crack remain. Further processing, such as only keeping a minimal length path, is performed to reduce noise. The approach achieved an F1 score of 83% in testing. The F1 score combines the measures of precision (the proportion of the retrieved instances that were relevant) and recall (the proportion of the relevant instances that were retrieved) to determine the performance of a classifier. Given that the approach relies on thresholding the cost of mapped out paths, there is no indication of how it would perform on images with no cracks present. Additionally, the approach requires that the images have no variance in lighting such as shadows, with an active lighting system being used for capturing images.

The approaches examined so far have shown susceptibility to noise causing false positive detections. An approach to overcome this is to initially filter out the noise. Fujita and Hamamoto (2010) have attempted this on concrete images, using a two-step process to detect defects with the first step being to remove noise. Their first processing step is to generate a corrective image by smoothing the original image with a medial filter. This

corrective image correlates with the noise in the image and is then subtracted from the original image. As shown in Figure 2-5, this effectively removes much of the noise, such as shadows, from the image. However, the cracks in the image in Figure 2-5, and all the images tested, are all clearly visible and greatly contrasting in intensity from the background, so the method might not be so successful in situations where this is not the case. Their second processing step used looks to then emphasise the varying width cracks in this reduced noise image through using a multi-scale Hessian matrix-based line filter. This is then followed by a process of probabilistic relaxation which looks at the probability of each pixel being a crack based on pixels neighbouring in straight lines at different angles, due to the thin shape expected of a crack. Finally, a locally adaptive threshold detects missing crack pixels to extend cracks more finely and close unconnected cracks.

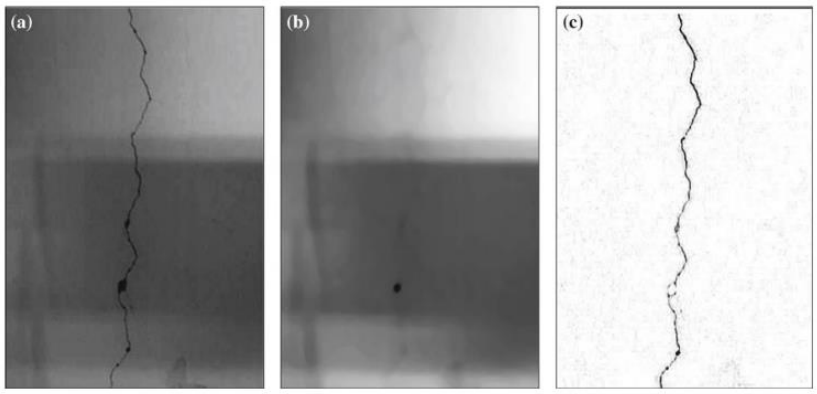


Figure 2-5: Median filter noise removal subtraction process: a) original image, b) corrective image (image smoothed with median filter), c) subtracted image (Fujita and Hamamoto, 2010)

When detecting defects in road pavements, the lane markings create noise and inhomogeneity in the images which could mistakenly be classified as a defect. Similar to the noise removal process described above, Nguyen et al. (2009) attempted to remove lane markings from images before detecting cracks in the remaining image. Here they use a Hough transform (Hough, 1960) to detect straight lines in an intensity-based threshold of the image being examined. The detected Hough lines then outline the lane marking regions which are then not considered in defect detection. A measure of anisotropy is then used to determine whether each pixel in an image is likely to be a crack. This works in the same way as the probabilistic relaxation method described above, in that where neighbouring pixels in a line along one orientation are very different to those in other orientations, it is likely to be a crack pixel due to the narrow nature of cracks. Joints and

bridged defects in road pavements are also linear, so they are also detected by the anisotropy measure. To differentiate between joints, bridged defects and cracks a machine learning approach has been used. A neural network with two hidden layers classifies images into the classes: defect free, crack, joint and bridged, based on features of the detected defect region and the surrounding image. Testing their method has shown good accuracy, though they acknowledge that some bridged defects are too similar to the background texture to detect. Additionally, data was collected using a platform that actively lights the surface, meaning that environmental lighting conditions have no impact on data quality.

Machine learning approaches are pattern based, looking to detect patterns in the features they are presented to make decisions, or classifications. These inputted features are manually determined based on what is specific about the data of interest relative to the background data, so that the different classes are separated in the feature space. Compared to manually choosing thresholds on features as has been described thus far, this has the advantage that classification is based on the data presented, so is more generalisable.

A machine learning classification is used by Oliveira and Correia (2013) to detect and classify cracks in road surfaces. First, they have performed pre-processing, involving pixel intensity normalisation to deal with non-uniform background illumination, and pixel saturation (replacing pixels brighter than mean intensity with the mean) to deal with bright spots due to reflective particles in the road surface. An unsupervised classification then determines the presence of a crack in non-overlapping segmented blocks of the input image, based on the average and standard deviation of the pixel intensity in each block. The expectation is that blocks with crack pixels will show higher standard deviation and lower mean pixel intensity values. They tested 6 different one class classifiers and clustering algorithms, finding that Gaussian Mixture Model clustering algorithm performed best with a 93.5% F1 score. It was found that detecting cracks less than 2mm thick resulted in many false positives due to the difficulty distinguishing cracking from other distresses. Additionally, they tested on images without shadows or wetness to avoid image noise, and as such cracks were clear, with obvious contrast in all images.

Jahanshahi et al. (2013) have used a machine learning based technique to detect cracks in concrete. Line shaped structuring elements at different angles are first used to highlight crack like patterns in the image. The structuring elements are tuned based on the crack thickness of interest and the scale of the image determined using Structure from Motion (SFM). Otsu’s method is then used to threshold the crack like patterns from the

background. Features of these segmented crack-like patterns, such as the ratio between the extracted area and its perimeter, are then calculated which are used by different classifiers to categorise the patterns as crack or non-crack. A 3-layer neural network, a Support Vector Machine and Nearest Neighbour classifiers are compared to assess their performance. The neural network classifier is found to perform best, and to outperform simple Canny edge detection. The method was tested on clean concrete images with clear well contrasted cracks. There were background objects in images, which the Canny method detected, but the proposed method was able to ignore.

A similar approach has been performed by Prasanna et al. (2016), also to detect cracking in concrete. They used a Random Sample Consensus (RANSAC) algorithm to fit line segments to pixels in the image likely to represent cracks (those where the intensity is below the average intensity by a predetermined amount). Features giving a quantitative description of the pixels in each line segment were calculated. A range of features are used based on intensity, gradient, and scale-space of the line segment pixels. The authors found experimentally that combining these multiple features (each providing weak cues) into one appearance vector gave optimal performance, compared to testing features individually. The features are used in a classifier to determine whether each line segment belongs to a crack or not. Support Vector Machine, Adaboost and Random Forest classifiers are used, with all three showing a similar validation performance. For this work, data was collected with robot mounted cameras on smooth surfaces of concrete bridges. There was therefore limited noise in the data. The classifier was trained and validated on a dataset with 2,000 15 x 15 px images from two bridges, balanced between crack and non-crack samples. The performance of the Adaboost classifier was similar when tested on data from a third bridge to on the validation data, though performance of the other two classifiers was significantly worse, suggesting they failed to generalise.

Shi et al. (2016) use a machine learning based process to learn the inherent structured information of cracks. Small image windows each depicting different features of cracks are used as a training dataset. Intensity, colour, and gradient based features from these are used to train a Random Structured Forest algorithm which is used to classify image patches into the type of crack feature they most closely represent. Edge detection followed by morphological operations are used for pixel-wise segmentation and joining of crack regions in these image patches. Further classification using a Structured Vector Machine is then used to separate false positives from crack regions.

The machine learning approaches to defect detection examined have the advantage over traditional hardcoded approaches, in that they can learn the inherent structural information of the defects being sought and are therefore much more capable of supressing image noise. However, in all the approaches examined using machine learning, the surfaces being classified showed little noise.

Compared to machine learning, deep learning has the advantage that it does not rely on handcrafted features from the data being examined for classification. Instead, the data itself is used for the classification, with features for classification learned automatically from the data. These features can therefore be far more abstract, meaning more information from the data can be used for classification. However, this also means that greater care is needed during training, with larger training datasets to ensure that the features learned are generalised and not just a nuance of the data used for training.

Several different studies have compared the performance of machine learning and deep learning approaches for detecting defects in concrete and asphalt pavement surfaces. These approaches include works by Abdelkader et al. (2020), L. Zhang et al. (2016) and A. Zhang et al. (2018). Abdelkader et al. (2020) proposes using singular value decomposition to reduce the dimensionality of image data for use in an Elman Recurrent Neural Network. They compared the performance for detecting multiple defect types in concrete, both to several machine learning and other deep learning approaches of various depths. The dataset used for training only consisted of 200 images, and as such the proposed method and a convolutional neural network, both relatively shallow networks only 2 layers deep performed best. Deeper convolutional neural networks performed worse than the machine learning approaches tested. L. Zhang et al. (2016) used a relatively shallow convolutional neural network with 4 convolutional layers to detect cracks in road pavement images. The model was trained on a dataset of 640,000 sample patches from 500 images and tested on a further 200,000 patches. The proposed method achieved a 90% F1 score which compares favourably with boosting and SVM machine learning methods which achieved 75% and 74% F1 scores respectively. A. Zhang et al. (2018) uses 2.5D image data from road pavements, where the depth of the image is included as well as the colour values in a Convolutional Neural Network (CNN) to detect the presence of cracking. They have smoothed the depth data to remove unevenness in the road pavement, so that the network can learn the depth features of cracks at a local scale without the effect of larger scale surface unevenness. A custom CNN was trained on a dataset of 2500 images and it was found to be more robust at eliminating local noises

and detecting finer, hairline cracks, giving better recall and precision when compared to a machine learning method with handcrafted features.

For image classification, Convolutional Neural Networks are generally considered the most effective deep learning models. They have the unique feature of convolution layers, which consist of learnable filters that are convolved across their input. This means they can take advantage of the local connectivity of an image, whereby features from different parts of an image can activate the same filter. This vastly reduces the number of parameters that need to be trained, when compared to a fully connected network, which enables the processing of larger images. In the work of Abdelkader et al. (2020) above, a recurrent neural network was used, which is fully connected, and therefore it was necessary to reduce the dimensionality of input images through singular value decomposition in order to compensate for the increased connectivity and hence number of parameters of this model.

Other examples using CNNs in the literature include works by Dorafshan, Thomas, and Maguire (2018), Cha et al. (2017), Li et al. (2018) and A. Zhang et al. (2018). Dorafshan, Thomas, and Maguire (2018) and Cha et al. (2017) both compare the accuracy of deep learning CNNs for detecting cracks in concrete surfaces with the feature driven edge detection methods that were examined initially. They find that the deep learning approaches are much less susceptible to noise in images, and as a result give a much more accurate classification. Cha et al. (2017) also looks at the effect of the training dataset size on the accuracy of classification. They have varied the training dataset size from 2,000 to 40,000 image window patches taken from 227 images. In this situation it is found that when training on more than 10,000 image patches, the validation accuracy does not improve significantly. However, the network being trained is only 4 convolutional layers deep, so has much fewer parameters to learn than would be the case on a deeper network where additional training data would be more beneficial. Li et al. (2018) detect cracks in concrete using a two-stage predictor with both stages using CNNs, based on the idea that whether a pixel is part of a crack is dependent on its context at both a local and a global scale. They found that predictions that additionally use global context information have fewer false positives when compared to only considering the local context, though the true positive rate is also reduced.

The CNN based approaches for concrete and road defect detection discussed so far have employed custom and generally quite shallow network architectures. It has been found that classification accuracy can be improved through deeper, more complex architectures,

but this results in increased training difficulty (He et al., 2016; Bianco et al., 2018). As a result, to avoid overfitting the data, which can cause a loss of generalisability, deeper networks require a larger training dataset. In some fields, large training datasets are readily available (Deng et al., 2009), but annotated datasets of infrastructure tend to be much smaller, and the creation of such datasets is a costly and manually intensive process. Transfer learning is a process through which neural networks can be trained on one domain where a lot of data is available and applied to another domain. This is either achieved either through using a pre-trained CNN as a feature extractor that feeds a classifier that is trained on the target domain, or by fine-tuning all the CNN weights on the target domain.

It has been observed in many deep learning networks that the initial layers learn more generic features which are not specific to a particular dataset or task. An example of such feature is a colour blob. By the last layer of the network, these features have transitioned to being specific to the task trained upon, for example recognising the shape of a nose. The transferability of these more specific learned features therefore decreases as the distance between the pre-trained task and the target task increases (Yosinski et al., 2014). Therefore, using a pretrained CNN as a feature extractor is most suitable where the target and initial domain are similar, and less data is available in the target domain compared to that needed for fine-tuning all the CNN weights.

Shin et al. (2016) examines the performance of transfer learning in the medical domain, where a large, annotated dataset is not available. They compare random gaussian parameter initialisations with parameters pre-trained on ImageNet data for 3 different CNNs of increasing complexity and depth. The training dataset consisted of 905 images from 120 patients. It was found that GoogLeNet, the deepest and most complex of the three CNNs tested, performed best when using transfer learning. However, when it was randomly initialised it overfitted the training data. Transfer learning was found to be consistently beneficial when compared to random initialisations despite the content of the ImageNet dataset that was used for pre-training being quite different from the target domain.

For detecting defects in non-masonry infrastructure assets, several approaches have applied transfer learning, either through use as a feature extractor or through fine-tuning a pre-trained network. These include works by Schmugge et al. (2016), Gopalakrishnan et al. (2017), Dorafshan et al. (2018), Hüthwohl and Brilakis (2018), Perez et al. (2019), Dorafshan, Thomas, and Maguire (2018) and Özgenel and Sorguç (2018), where the use

of transfer learning has enabled the authors to accurately train deep CNNs with a limited amount of data. Schmugge et al. (2016) and Dorafshan et al. (2018) both compared the results of using a deeper network with a fine-tuning transfer learning approach, to using a less deep network with a random initialisation. They found that transfer learning resulted in a significant increase in performance, especially in noisy regions. It was also suggested that the transfer learning approach resulted in a classifier that was more able to generalise, not being thrown off as readily by edges of concrete, or vegetation not seen in the training dataset. Dorafshan, Thomas, and Maguire (2018) compared the performance of fine-tuning and feature extractor transfer learning approaches for detecting cracks in concrete surfaces using an AlexNet classifier. They found that the fine-tuning approach gave the highest accuracy, though both approaches performed better than fully training the network from a random initialisation. Özgenel and Sorguç (2018) have looked at the effect of the number of training samples required to fine-tune several of the leading deep CNNs pre-trained on ImageNet and have found that the best accuracy was achieved when more than 14,000 training samples were used.

It is often observed that there are more instances of negative samples, where the defect sought is not present, than of positive samples. This is especially the case with defects in infrastructure assets, where there is usually a lot more good condition surface than surface with defects visible. It is known that a class imbalance will cause a reduction in the error of the majority class, but an increase in the minority class, as the majority class will dominate the gradients when updating parameters (Anand et al., 1993). In infrastructure assets, this will result in an increase in the error of the defect class as the minority class. Shin et al. (2016) observe that when trained with a biased dataset, the classification accuracy results of a GoogLeNet classifier are lower than when trained with a balanced dataset. Guo et al. (2020) look at different methods for dealing with class imbalance in a dataset of defects in building facades, where one defect class was just 1% of the data samples. The methods considered are oversampling the classes with fewer samples, a weighted loss function to counter the class imbalance, and a meta-learning approach where weights were reassigned based on the gradient directions of a balanced validation set during training. It was found that all the approaches outperformed training with imbalanced classes. The meta-learning approach performed best.

Using a CNN as a classifier gives a very course, block-based classification, based on the size of image windows used. To determine within each block where a defect is located, one approach that has been considered is Class Activation Mapping (CAM). This utilises

the ability of CNNs to localise the objects causing each class categorisation in the convolution layers, despite just being trained on class labels and not class locations. This localisation is usually lost in the fully connected layers used for generating the classification, but CAM uses the gradients from the final convolutional layer, before the fully connected layers of the trained network, to indicate the image regions which cause the given class to be assigned (Zhou et al., 2016). CAM has been used successfully by Perez et al. (2019) and Guo et al. (2020) to accurately locate defect locations in classifications of images of buildings. However, in some cases in these works, the CAM location has been found to be incorrect, suggesting that other features of the image have given rise to the given class. This highlights CAM's ability to help to understand the features of images the CNN is using to decide on a classification, helping to determine how well the CNN has been trained.

Aside from CAM, other examples of semantically localising the defective pixels are shown in the work of Dorafshan, Thomas, and Maguire (2018), Alipour et al. (2019), Yang et al. (2018) and Liu et al. (2019). Dorafshan, Thomas, and Maguire (2018) use an edge detection algorithm to identify the defective pixels within the image patches of concrete surfaces that are classified as cracking by the CNN. Alipour et al. (2019), Yang et al. (2018) and Liu et al. (2019) however perform the semantic segmentation for crack detection within the neural network. This is done using a fully convolutional network, where the fully connected layers traditionally seen at the end of a CNN, responsible for losing the localisation ability of the network, are replaced with transposed convolutional layers which up-sample the prediction heatmaps to the size of the input images. Additionally, skip connections are used to connect the convolutional layers and transposed convolutional layers at different depths, such that features of multiple scales can be fused in the output. This therefore makes the process less dependent on the image scale. Alipour et al. (2019) compare this approach to adaptive thresholding and Canny edge detection and shows it to be less susceptible to image noise, though Yang et al. (2018) suggest that when using this approach, the accuracy of crack segmentation near the image borders declines.

Koch et al. (2015) reviewed many proposed approaches for automated visual inspection of infrastructure and concluded that the computer-vision based methods they have examined are able to contribute successfully to the automation of the detection and measurements of defects in reinforced concrete. However, they identify that for bridges. most concrete crack detection approaches have focused on flat, simple areas, such as the

bridge deck, as opposed to joints and other areas where it is harder to distinguish cracks from true edges. Additionally, they identify existing methods need improvement as their performance degrades with noisy data and non-ideal environmental conditions, especially with shadowing. Rose et al. (2014), who have also reviewed proposed automated visual inspection approaches support this view, suggesting that none of the approaches they have examined could provide an acceptable automatic solution for crack detection and assessment. A more recent review, has also suggested that there is still a lack of evidence of performance of the examined methodologies in the field, with most work still being lab experiments where noise is more controlled (Agnisarman et al., 2019). As most of the approaches examined here are machine and deep learning approaches, this suggests their performance with the noisier surface and environmental conditions would be questionable due to the lack of generalisation with that data.

As discussed above, hardcoded and machine learning approaches such as the works by Fujita and Hamamoto (2010), Nguyen et al. (2009) and Oliveira and Correia (2013) have sought to reduce the impact of noise in the data through pre-processing the data to remove the noisy regions or reduce the effect of the noise. However, with deep learning, more researchers have attempted to teach the classifier to recognise the difference between noise and true positive data. This is shown in the work of Schmugge et al. (2016), where images of concrete from a nuclear power plant are examined for cracking. These images were low contrast and contained features such as welds and scratches with similar appearances to cracks. When building the training dataset, crack free patches with more noise were deliberately chosen to expose the network to these features so it could learn to differentiate. Additionally, the methodology was tested on video data which meant that neighbouring video frames could be used to correct false positive and negative classifications. The converse is shown in the work of Dorafshan et al. (2018) where training was performed on photos of lab made concrete bridge decks. When subjected to datasets with more image noise they found the classification performance degraded. Özgenel and Sorguç (2018) therefore suggests that the level of variance within a training dataset is more important than the number of samples for training an accurate classifier.

As has been discussed, compared to concrete and asphalt surfaces, masonry surfaces present much more noise to a classifier due to the brick-mortar interfaces and the shadows caused by mortar joints, which have the potential to appear like defects. Formwork lines on concrete surfaces, though not as pronounced as mortar joints in masonry, also have potential to cause such confusion, but Hüthwohl and Brilakis (2018) suggest that their

model is able to differentiate between formwork lines and cracks, despite the formwork lines having greater footprint in profile. On the other hand, Gopalakrishnan et al. (2017) suggest based on their misclassified crack images, that their classifier has failed to learn to distinguish between cracks and joints in concrete pavements: joints are deeper and more akin to mortar joints than formwork marks.

Alipour and Harris (2020) identify that surface cracks in dissimilar materials have a fundamental similarity in that they present a discontinuity of the material. They have therefore looked at the suitability of classifiers trained on one material to be applied to another for detecting cracks. They found that a network trained on concrete surfaces has an accuracy of 97% on concrete, but only a 60% accuracy on asphalt. Conversely, a network trained on asphalt has 88% accuracy on asphalt, but 57% accuracy on concrete. This drop in performance shows that differences in other surface features between concrete and asphalt surfaces are significant. Özgenel and Sorguç (2018) have performed a similar study looking at the transferability of learned features from CNNs trained to detect cracks in concrete building facades by testing on other textures, including masonry as well as pavements and building components. They suggest that the features learned during training are transferrable to other materials with high accuracy, though note that masonry is the most challenging among their test cases due to the brickwork jointing and background textures. Choi and Cha (2020) have developed a semantic crack detection CNN for images of concrete with noisy background features. They have shown that when training on an unvarying concrete dataset, classification performance on a noisy concrete dataset is significantly worse than on the unvarying dataset. However, when training on the noisy dataset, there is a much smaller degradation in testing performance between the noisy and unvarying dataset. As a result of this, better masonry classification performance is achieved with classifiers optimised for masonry, especially where the masonry images are noisy. The existing work on defect detection in masonry is discussed below.

2.3.2 Masonry defect detection

Work to identify defects in masonry has looked at exploiting both geometric and visual cues. In the following paragraphs, geometric methods are discussed first, followed by visual methods.

Quagliarini et al. (2017), Chen et al. (2019), Hallermann and Morgenthal (2016) and Galantucci and Fatiguso (2019) all looked to manually exploit geometric cues in point clouds depicting masonry, created with laser scanning and photogrammetry. It has been

shown that by visualising the deviations in an uncoloured 3D model, areas suffering from material loss can be identified. Additionally, planes can be fitted to building elements and the deviation of each scan point measured, highlighting areas of material loss, which can then be compared with photographic and historical data. These areas of deviation can be measured in the point cloud to quantify damage. Cracks can also be extracted by using a gradient filter to detect edges in the derived deviation map, though in the case-study of Galantucci and Fatiguso (2019), some mortar joints were incorrectly classified as cracks and areas of material loss. Ye et al. (2018) used a similar approach for measuring distortions in masonry arch bridges. They have fitted primitive cylinders to the point clouds of arch barrels to measure their deviations from their design intent, to determine the most likely movement scenarios. These can be validated by examining the location of visible defects on the bridge. Sánchez-Aparicio et al. (2018) combined the above two approaches, detecting both deformations and material loss. For finding areas of material loss, rather than relying on deviations from a fitted plane as in previous examples, a CANUPO (Caractérisation de Nuages de Points) classifier (Brodu and Lague, 2012) has been trained. CANUPO is a machine learning classification algorithm used to discriminate between different features in point clouds.

Machine learning approaches have also been used for classifying the visual cues in masonry. These approaches have focused on heritage rubble and ashlar masonry building conservation. They have used unsupervised clustering algorithms for segmenting image pixels based on intensity into both the materials present and the types of damage present (Meroño et al. 2015, Sánchez-Aparicio et al. 2018, Del Pozo et al. 2016, Armesto-González et al. 2010). Examples of materials segmented are mortar, wood, stone and iron, and examples of damages are biological, cracking, moisture, and salts. In these works, visual condition data has been acquired either through photographic means or the radiometric output of laser scanning, i.e., the intensity of reflections. The effect of the wavelengths of light captured on the ability to differentiate between the spectral signatures of different surfaces in images has been examined with the use of multi spectral cameras, with a suggestion that the inclusion of the near infrared as well as visible wavelengths increased the degree of recognition among different damages (Meroño et al. 2015, Armesto-González et al. 2010). However, Del Pozo et al. (2016) suggest more of a difference is made by using active sensors, such as on a laser scanner where the reflected response from an active source is measured, as opposed to passive sensors, such as a

camera. This is because these sensors are not affected by changing lighting conditions and shading.

It has been suggested that with heritage building conservation, the current standards for diagnosis lack quantitative criteria to identify and classify damage and determine severity, and there is therefore a need to develop such standards (Galantucci et al., 2018). The above works have demonstrated the ability to extract such information from both geometric and visual data, though in each case the works are case studies, tested only on the structures they were developed with, suggesting they may not be generalisable to other structures and datasets.

Another method used to detect defects in masonry infrastructure is to detect changes in the structure from one data capture to the next. This is demonstrated in the work of Mett and Eder (2019) and McCormick et al. (2014) for detecting defects in tunnels. Here models made with both photographic image data and laser scanner captured geometry data are compared with historic data by aligning the models and using Digital Image Correlation (DIC). This can highlight changes down to as little as 1mm between inspections. However, this can only highlight changes from a previous recording of the structure, so pre-existing defects will not be detected. Additionally, the examples showing successful detection of changes were for those larger changes, for example the removal of a patch of soot from a tunnel wall, or footprints in rail track ballast. DIC coefficient maps contained a lot of noise, meaning smaller changes such as cracking may be less obvious. On bridges where conditions such as lighting are harder to control when compared to tunnels, the level of background noise in DIC maps is likely to be much higher. DIC has been used to successfully detect cracking in lab experiments of loaded brick samples (Tung et al., 2008), and in the field to capture the full-field deformation of a bridge between a loaded and an unloaded state (McCormick and Lord, 2012). In these works, it is possible to see the strains of a crack in the coefficient map as a displacement change, though for the case of the bridge the crack was known about prior to imaging and therefore had been focused on for imaging at high resolution.

As with concrete and asphalt infrastructure, methods developed for automatically detecting defects in masonry images have varied between using hardcoded methods, using machine learning with manually handcrafted features, and using deep learning. Hardcoded methods have been examined by McRobbie (2008) for detecting defects in masonry highway bridges. They have investigated using a Haar transform and image entropy to segment defects from the bridge background. Whilst the method worked with

concrete bridges, they found the bricks and mortar in masonry tended to swamp out any defects from the detection. Ellenberg et al. (2014) examined a combination of edge detection, percolation, colour standard deviation and Hough transform approaches for detecting cracks in images of masonry, yet the method performed poorly. Similarly, the authors identify that the main challenge for detection is noise, such as that caused by surface roughness and the patterns of the masonry. They suggest a potential approach for overcoming this is to filter out inherent noisy parts of the structure from the baseline pictures, though acknowledge the problem with such an approach is that, particularly with masonry, most of the damage initially occurs around the noisy features such as the mortar joints. In effect Schmitt et al. (2000) does this by attempting to detect defects in individual, isolated bricks as part of the manufacturing process. Their method identifies and measures cracks, surface indentations and protrusions, and discolorations. This is done through edge detection, thresholding, and skeletonising processes, and can accurately detect defects in the isolated bricks themselves when no mortar interface is present.

Valero, Forster, et al. (2019) and Wang et al. (2018) have filtered the noisy mortar joints from the structure before attempting to detect defects as suggested above by Ellenberg et al. (2014). Each brick has therefore been inspected for defects individually. In the work of Valero, Forster, et al. (2019), the coloured 2.5D surface of ashlar masonry blocks is acquired by both laser scanning and photography, and features are extracted for each stone. Features include the roughness of the face and hue value standard deviation, to determine the likelihood of each stone being defective. The fine dressing of ashlar masonry facilitates this process. Defective areas within likely defective stones are then determined through fitting a plane to the depth map and identifying outlying points as well as identifying points whose colour values vary significantly from the average for that stone. These defective areas are then classified using a trained machine learning classifier based on a logistic regression model using both geometric and colour features. The authors identify that more data is required to test their method for generalisability as it has been tested on the same building surface it was trained on. Wang et al. (2018) use a CNN-based classifier to classify each brick in a masonry wall based on the presence of cracking, efflorescence, and spalling damage. Both AlexNet and GoogLeNet classifiers are used, and trained on datasets of both 2,000 brick samples, and 20,000 brick samples (including data augmentation). AlexNet validation accuracy increases from 93% to 93.3% with the larger training dataset, whilst GoogLeNet increases from 93.3% to 94.3%. This

suggests the larger and deeper GoogLeNet model is better able to learn the larger dataset, although the difference is relatively small. As before, this study does not test the generalisation of the models, with validation performed on the same structure as training.

Other approaches have foregone prior segmentation of mortar and brick regions, instead directly detecting defects in the inhomogeneous masonry. These have almost exclusively used CNN-based approaches, except for Samy et al. (2016) which has used a machine learning approach with a SVM classifier. They have used 13 features incorporating both textural and pixel intensity data of 2.5D images taken with a Microsoft Kinect 2 sensor. These are used by the classifier to detect damage on bricks, mortar loss and cracking. The method was able to correctly identify 100% of the defect classes, though 16.7% of the clean images were falsely classified as defective. In this case the training and testing datasets were both captured from a lab-built masonry wall, so there is no evidence of generalisability. Additionally, the images were therefore taken under well controlled conditions, and the defects were well defined with minimal image noise. Hallee et al. (2021) compare the ability of CNNs to detect cracking in masonry between lab-controlled masonry images and real-world masonry. Training was on image window patches from a dataset consisting of 53 lab-built and cracked brick walls. Meanwhile, the method was tested on real world masonry images acquired online as well as image patches from the lab-built brick walls. CNNs with depths of 10 and 18 layers were compared. The shallower CNN performed better on the lab-based data with an F1 score of 91.5% compared to 90.5%. However, its generalisation to real world data was much worse, with its F1 score decreasing to 62.8% compared to 82.4% for the deeper CNN. This suggests that the additional trainable parameters of the deeper model provide an advantage in domain adaption, suggesting the extra filters can provide additional information to the classifier. Additionally, this shows that reliable performance on homogeneous lab-based data does not translate to real world performance.

Another approach for damage detection in masonry is proposed by Wang et al. (2019), who use a region based convolutional neural network (R-CNN) to identify and locate spalling and efflorescence in masonry images. This work also showed a lack of generalisation of approach, with all image data taken from the same structure. Marin et al. (2021) have similarly used an R-CNN to successfully detect and locate cracks on images of different types of masonry. They have employed a system of progressive detection whereby more detailed images are only processed if defects are detected in a

general view. Such a system is applicable in this case, as the cracks being sought are all quite large, so discernible from further afield.

As with concrete and asphalt infrastructure, work has been done to semantically locate defects in masonry structures. Dais et al. (2021) used a dataset of masonry from buildings which also contain background distractions such as windows and pipes. The dataset also varied in terms of age and colour of masonry units, with both clay and concrete blocks included. The work first classifies image window patches into crack/non crack with an accuracy of 95.3% using a fine-tuned MobileNet CNN pre-trained on ImageNet. Transfer learning was found to outperform a random initialisation. Additionally, an improvement in precision was found when extra types of background objects were included in the training dataset as the network learnt to ignore them. Semantic segmentation was then performed, with U-Net and Feature Pyramid Network (FPN) architectures found to perform best, with F1 scores of 79.6%. Again, a worse performance has been obtained for non-pretrained networks, where the F1 score declines to 75.4%. In this work, the cracks being sought in the dataset were all large and clearly defined against their masonry background. The transferability of the proposed network trained on masonry surfaces, is examined by testing on concrete surfaces where its F1 score declines to 74.7%. This is much less of a deterioration when compared to a network trained on concrete and tested on masonry as reported above. This is as expected since masonry surfaces are more complicated than concrete.

Chaiyasarn, Sharma, et al., (2018) and Wang et al. (2018) are the only examples found in the literature where the masonry condition is of a similar noise level to that seen on masonry arch bridges. All other works examined have focused either on lab curated masonry, or masonry from the facades of modern or well-preserved heritage buildings, which therefore have much more well-defined defects. Chaiyasarn, Sharma, et al. (2018) use a CNN to classify the presence of cracking in image windows. They have used a dataset of 3601 crack patches for training, giving a validation accuracy of 86% and testing accuracy of 74%. In the dataset for this study, only large and more well-defined cracks are included. It was found that in images there were many false detection areas, particularly around the mortar joints, suggesting that mortar joints were being confused with cracks. Wang et al. (2018) also draw a similar conclusion suggesting that defect detection in masonry structures is difficult because of the similar appearance between mortar joints and defects. They have used a R-CNN approach to search for missing and heavily spalled masonry units.

2.4 Mortar Joint detection

2.4.1 Motivation for detection of mortar joints

As discussed previously, several attempts at detecting defects in masonry surfaces have reported that the similar appearance between mortar joints and defects in masonry surface has led to false detections of defects around mortar areas. Additionally, several of the other discussed attempts to detect defects in masonry surfaces have initially pre-separated the brick regions from the masonry. This simplifies the heterogeneous nature of masonry, making detecting masonry defects much more comparable to concrete, where a lot more of the existing work has focused. If transferring those learned features from concrete to masonry surfaces, it is known that the transferability of features in deep learning networks decreases as the difference between the base and target task increases (Yosinski et al., 2014). For separating masonry surfaces into mortar and brick regions, it is important to accurately identify the locations of the mortar joints.

As well as potentially aiding the detection of defects in masonry surfaces, identifying the location of mortar joints can also be used alone as an indication of the state of masonry bridges. This is because features and distortions in the bonding pattern can be indications of distortions in the bridge. For example, distortions in the bedding plane could be indications of differential settlement of the bridge, giving a better indication of long term movement than measuring cracks which may have been repointed (McKibbins et al., 2006).

Mortar joint locations can also be used to quantify the size of damaged areas, and therefore the material requirement for a repair. In heritage masonry preservation, there is also a need to label the condition of each stone individually (Forster et al., 2020), with previous work showing the ability to extract parameters such as the colour and face profile for each stone individually for documentation (Valero, Bosché, et al., 2019). Other work has demonstrated the utility of segmenting mortar joints when creating Historic Building Information Models (HBIM) of masonry structures as it enables a hierarchical configuration of information by layering the point cloud, the 3D model, the segmented masonry units, and the defective masonry units (Valero, Forster, et al., 2018). Knowing the location of mortar joints within 3D models of structures means the recess of the mortar can be accurately measured to detect recessed zones and estimate the amount of repointing required. The depth of the joint recess also highlights areas of masonry that

may be vulnerable to progressive loosening of material if left unattended (Valero, Bosché, et al., 2018).

With an inhomogeneous material such as masonry, knowing the arrangement of the different constituents can help to understand how a structure will respond to loading. Masonry subjected to loading in laboratory tests has shown much larger strains in mortar areas than brick areas, due to the higher stiffness of brick (Tung et al., 2008). Lourenço (2002) suggests that for estimating the load carrying capacity of historic masonry structures, micro-modelling approaches are best, due to the importance of the masonry unit to mortar interface on the structural behaviour. The mortar bond is often the weakest link in masonry, giving a non-linear response, which is one of the most relevant features of masonry's behaviour. They have compared micro and macro modelling methods for a simple masonry arch, finding that the different approaches lead to different safety factors as well as different load displacement diagrams. Micro-modelling approaches for masonry look to model a discontinuum at the mortar interface, whereas macro-modelling smear the mortar and interface into a homogeneous continuum.

Napolitano and Glisic (2019) has also shown how through micro-modelling masonry structures, it is possible to predict the loading that has caused crack patterns. They have experimentally loaded masonry walls to induce cracking. These have been modelled with several different loading cases, including the actual loading case. It was found that both quantitatively and qualitatively, the simulated crack pattern most closely resembling the experimental one was the real loading case. Even for macro-modelling approaches, knowing the position and size of mortar joints has shown to be beneficial. In the work of Cluni et al. (2015) and Cavalagli et al. (2013), images of masonry are separated into mortar and brick pixels to calculate the masonry unit sizes and the mortar thicknesses, which are then used to estimate the elastic properties of the masonry continuum.

Several different studies have compared the effectiveness of micro-modelling masonry arch bridge like structures. In the work of Kassotakis et al. (2021), 25 square arch cross sections have been built both physically and numerically with the same span, rise, and thickness, but different bed joint location detail. It was found that the joint position, especially the inclination angle was found to influence the collapse load directly and linearly. This suggests that when developing models of arch particular care must be taken to accurately obtain the geometry of the bed joints. This conclusion agrees with previous work by Y. Zhang et al. (2016), who suggest that for square arches, modelling of the bed joints is much more important than the head joints. For this, they have modelled arches

as a series of solid elements smeared through the thickness of the arch and found it to provide an accurate result compared to fully micro-modelling the arch. With arch bridges, ring separation can lead to a big reduction in their strength. Y. Zhang et al. (2018) has found that where mortar joints are weak, different masonry bonding patterns can have a large effect on the stiffness and ultimate load of the arch, as the bonding pattern can influence the onset of ring separation. For skew arch bridges, Forgács et al. (2017) have found that both the dimensions of the individual masonry units as well as their bonding pattern affect the bridges capacity. This is a result of different bonding patterns giving rise to different amounts of friction between the masonry units, where for instance having a zigzag at the hinge increases friction as opposed to bricks being straight to the hinge. As a result, for understanding masonry arch bridges, the use of detailed micro-scale models representing the actual masonry bonding pattern is critical, as reduced models may provide an incomplete picture (Y. Zhang et al., 2018).

It has also been shown to be advantageous to model the exact geometry of the masonry in structures, rather than just idealising the size and shape of the individual masonry units and mortar joints. Work by Loverdos et al. (2021) has micro-modelled masonry structures based on the exact geometry of the individual masonry units and mortar from image data and has compared this approach to using an idealised masonry geometry in the model. On a model of a masonry arch doorway, it was found that the maximum loading of the geometrically accurate model was 9.7% greater than that for the idealised one. Additionally, a cracked brick wall has been modelled, assuming zero strength in the crack. Here, the geometrically accurate masonry geometry had a 3% higher maximum loading than the idealised masonry geometry. Similar work on masonry arch bridges has been done by Riveiro et al. (2011) where a 3D micro-model of a bridge has been developed from a photogrammetric model, based on the exact geometry of the individual stones. They have found that the failure load in the model with the accurate, exact masonry geometry is about 12% higher than that for the idealised masonry geometry. Whereas the accurate model of the individual stones contours here has been obtained manually, Loverdos et al. (2021) has shown this possible to automate. For this, their method takes an image of masonry that has been binarised into its mortar and brick constituents as an input, which then acts as markers for a watershed transform. This then isolates and uniquely identifies individual regions in the image, which are then outlined using a border following algorithm, therefore outlining the individual bricks for use in a model. It has been suggested that as well as providing more accurate results, automatically micro-

modelling masonry structures in this way could save considerable effort compared to creating a model manually through idealised masonry geometries.

In summary, the motivation for detecting mortar joints in masonry arch bridges is:

- Mortar joint location could aid defect detection through simplifying heterogeneous masonry surface
- Mortar joint distortions can be useful for highlighting historic settlement and deformations of masonry arch bridges
- Mortar joint locations help quantify sized of damaged areas for estimating material requirements for repair
- Mortar joint locations can improve the accuracy of modelling bridges for calculation of their service capacity.

2.4.2 Methods of mortar joint detection

For these reasons, previous studies have looked at ways of detecting mortar joints in masonry surfaces. Within the field of heritage masonry preservation, several works have looked at the ability to segment mortar regions alongside other damage types within intensity images (Del Pozo et al., 2016; Armesto-González et al., 2010). Both works have used an unsupervised machine learning clustering classifier to group pixels based on their intensity, with two of the classes being mortar and stone. It is found that the ability to distinguish between stone and mortar is poor, and worse than distinguishing defect classes, due to the small thicknesses of mortar and the similar spectral signatures of stone and mortar. In the work of Yuan et al. (2020), building facades are classified into the materials present. This classification uses the reflectance intensity and colour and the geometry outputs of laser scanning with an Ensemble classifier. This is supervised machine learning, using features such as the surface roughness and material reflectance to learn and classify materials. Of all the building façade materials learned, mortar classification was found to have the lowest performance due to being misclassified as stone. The mortar surface class was also found to have the highest dispersion of multiple feature values, especially its surface roughness. Use of the reflectance intensity data from laser scanning also has the potential for inaccuracy and misclassification, as it is found that the presence of moisture can change the reflected intensity from a surface (Hassan et al., 2017).

Other methods for detecting mortar in masonry exploit geometric distortions that are expected at the mortar joints, rather than simply trying to classify the mortar texture and reflectance as above. Valero, Bosché, et al. (2018) use a Continuous Wavelet Transform (CWT) on a 2.5D depth map of rubble masonry surfaces. The CWT is tuned with a mother wavelet to the width of a mortar channel on the surface being examined so that it responds most strongly to the mortar regions, and not to waviness or curviness in the wall as the frequency for these is much lower. This gives an approximate segmentation of individual stones and mortar regions, though concave irregularities in stones cause some areas to be falsely labelled as mortar. To correct this, a convexity operator is applied to each stone segment, followed by iterative dilation. This work has been released as a plugin for CloudCompare (Girardeau-Montaut, 2014), which also contains a tool for manually labelling and correcting errors of the automatically generated mortar segmentation, suggesting some manual intervention is still required to satisfactory segment mortar (Enrique Valero et al., 2020).

While the above work focused on irregular stone masonry, which geometrically has well defined and deep mortar joints, the authors have also tried to expand the method to work with ashlar masonry, exhibiting much shallower mortar joints (Valero, Forster, et al., 2019). For this, both geometric and photographic data are used with a similar CWT based technique, though the authors give little information on the level of success, or manual intervention required to satisfactorily segment the mortar. Additionally, the method has only been applied to a single wall, so there is little evidence of generalisability. A similar technique looking to detect deformations in both point cloud and colour data to detect mortar joint locations has been developed by Sithole (2008). This is based on a technique of weighted proximity segmentation, whereby each point in the point cloud has its geometric and radiometric information compared with its neighbours to determine the strength of connected points, therefore finding connected regions which separately represent the mortar and brick regions. However, this method was most successful only when the mortar channel is deep enough to cause a surface discontinuity.

Some works have only exploited the expected difference in colour between the brick and mortar regions, not considering geometry. Hui et al. (2014) developed a technique for outlining individual bricks to count them and estimate the progress of construction. The application involves pristine condition masonry, with no dirt or other discolourations and distractors. Therefore, a colour thresholding technique can distinguish between the mortar and brick colours, followed by erosion and dilation operations to clean up the

segmentation. However, the authors recognise that the method has only been tested on masonry with red bricks and yellow mortar, so may not be effective on other masonry. Cluni et al. (2015) and Cavalagli et al. (2013) have also used thresholding to segment mortar, in this case thresholding the intensity of infrared images. Otsu's method (Otsu, 1979) is used to identify an appropriate threshold level for the segmentation. For this the image is divided into sub images to increase the uniformity of luminance in the segmentation. A similar approach is performed in the work of Oses et al. (2014) to segment greyscale masonry images. As before, the image is divided into smaller regions each about the size of one block and processed independently. Processing initially involves removing noise through removing pixel intensities with low frequency of occurrence. The region is then segmented into pixels likely to be mortar and brick respectively, through identifying intensities with high frequency of occurrence, using the assumption that the mortar will appear darker.

Other methods have just used the image data to detect mortar joints in masonry, utilising the known pattern of masonry blocks. This is exemplified in the work of Wang et al. (2018), where a sliding-window method is used to segment individual bricks from masonry. In this case the brick course height, header length and stretcher length and pattern are all known, so the sliding window progresses at known intervals from an initial position aligned on a single brick. The image has been rectified to create an orthophoto, and the imaged masonry is very tightly bonded, finely crafted masonry, meaning that all dimensions remain constant throughout the image. In most masonry structures, this regularity cannot be relied on, so other techniques are more involved. Riveiro et al. (2016) detect the position of joints in masonry, through exploiting the regular arrangement of masonry together with intensity discontinuities expected at the joints. Intensity discontinuities parallel to the bed joints are highlighted using a Sobel operator, and peaks in the sum of the outputted pixel intensities from each row of the image used to determine the bed joint locations. This process assumes that the image is orientated such that the bed joints run perfectly horizontally. The image is then segmented along these bed joints into individual brick courses and the same process is used to detect head joint locations. The determined locations of the bed and head joints are then used to define the centre of each brick, which is used as the initial markers for a Watershed function, to give the outline of each brick.

All the above mortar segmentation methods are hand-crafted to exploit features present in masonry. In the work of Ibrahim et al. (2019) a deep learning approach is examined. A

U-Net based neural network is trained and tested on a dataset of 117 and 32 annotated masonry images, respectively. These images include both rubble and ashlar masonry, though all the images the authors presented showed clean masonry with a good contrast against the mortar joints. The U-Net output is used to define the centre of each brick as an input to a Watershed algorithm, defining each brick's outline. The method showed a very good ability to segment bricks in the domain of the tested dataset, with an F1 score of 81.57. Performance was also shown to be superior, and less susceptible to noise than with hand-crafted methods. However, the masonry imaged in the dataset for this study was all much cleaner and less noisy than is the case for that found on masonry arch bridges, where mortar joints can be occluded by graffiti, paint, vegetation, and layers of dirt accumulated over their service life.

2.5 Chapter summary

From the literature reviewed, the need to monitor the condition of our historic stock of masonry arch bridges is apparent due to the role they play in underpinning much of our transportation infrastructure. The current methods for monitoring these bridges have been examined, especially regarding the visual inspection process, and the types of defects that are relevant to an inspection engineer. Additionally, the current deficiencies of visual inspections have been discussed. These include the danger, cost and disruption caused by inspections, the relative infrequency of inspections, and most importantly the subjectivity of inspections.

One key tool to minimise the consequence of subjective visual inspections is the use of photographs to capture parts of the structure. This provides a permanent visual record of the structure and means that if defects are subsequently discovered, historic photographs can be examined to determine the history of such a defect. This however relies on photographs having been taken of that part of the structure in the past, which with current visual inspection procedures is unlikely as photographs are focused on pre-acknowledged defects. As a result, there has been a significant effort to automatically capture image and geometry data of infrastructure using cameras and laser scanners mounted on UAVs, vehicles, and fixed pan-tilt units. These can provide detailed 3D models of infrastructure. For visualising these models in 2D the widespread practice is to take orthophotos from the model which works for planar, flat infrastructure. However, taking orthophotos of arch barrels results in compression of the image towards the spring line of the arch.

Subjectivity is still present in the manual interpretation of digital data, so numerous attempts have been made to automate the detection of defects within geometry and image data. This effort has primarily focused on asphalt and concrete infrastructure which generally is more homogenous and less noisy than masonry infrastructure. Different approaches can be categorised into hardcoded techniques, change detection techniques, machine learning techniques and deep learning techniques. Generally, it was found that the deep learning techniques were least susceptible to confusion by noisy images. Transfer learning was a technique used to improve the accuracy and generalisability of deep learning techniques when trained on limited datasets as is the case with annotated images of infrastructure. For defect detection in masonry, fewer studies have been conducted, the majority of which have considered relatively clean masonry. These studies have suggested that there is a level of confusion in detection between mortar joints and defects.

Therefore, detection of mortar joints in masonry could potentially help to improve detection of defects. Mortar joint detection also can be useful for highlighting historic settlement and deformations of masonry arch bridges, as well as their modelling for calculating their service capacity. As with defect detection, mortar joint detection approaches can be categorised into hardcoded techniques, machine learning techniques and deep learning techniques, and it is also found that the deep learning techniques are least susceptible to noise. However, none of the existing mortar joint detection techniques have been developed with the same level of masonry noise and distractions as seen on masonry arch bridges.

Therefore, this work has identified the following gaps in the literature which it looks to fill:

- Develop a methodology to visualise a 3D model of an arch barrel in 2D without causing compression and scaling of parts of the image
- Develop a methodology to automatically detect mortar joints in the noisy masonry surfaces of masonry arch bridges
- Determine the effect of mortar joints on the detection of defects in masonry surfaces
- Develop a state-of-the-art automated defect detection methodology which has better performance than current manual visual inspections.

3 DEFECT CLASS ANALYSIS

The work of this PhD looks to develop a methodology for automatically determining faults with masonry arch bridges from visual data. The outline of the proposed methodology is presented in Figure 3-1. This workflow is divided into three sections.

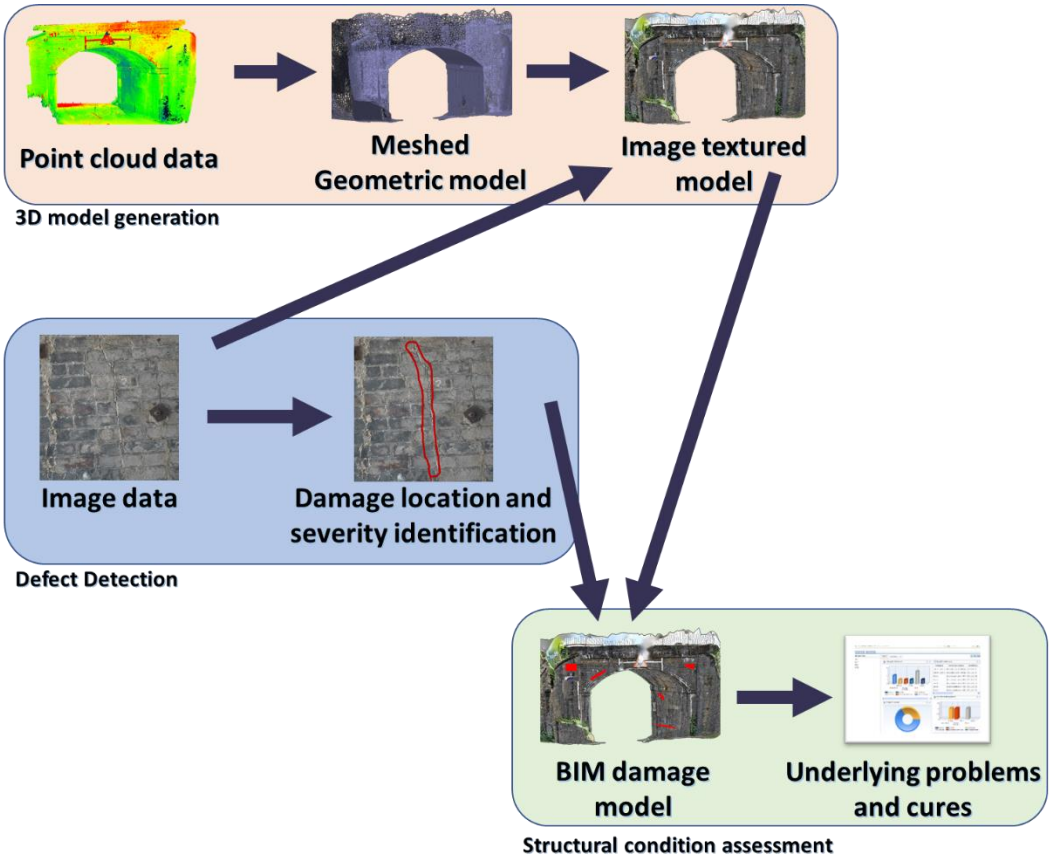


Figure 3-1: Methodology for automated masonry arch bridge assessment

The first section, highlighted in orange, looks to build image textured 3D models of masonry arch bridges. Here geometry data in the form of point clouds acquired either through laser scanning or photogrammetry is meshed to create a 3D model textured with high resolution surface imagery acquired through photography. This first section is not the primary contribution of this PhD, since, as discussed previously in the literature review, there are several different technologies that are capable of automatically generating image textured models of bridges. In this work, commercial tools have been used to manually generate image textured 3D models following the steps presented in the methodology in Chapter 4.

The second section, highlighted in blue, focuses on automatic detection of defects in images of masonry arch bridges. This is the primary focus of this PhD. To know what defect like features are worthwhile identifying, understanding of the nature of visible defects present in masonry arch bridges and their significance to the performance of the bridge is required.

The third section, highlighted in green, identifies these defects. To achieve this, the proposed workflow maps the detected defects onto the image textured model to create a BIM model with defect locations layered onto the geometry of the bridge. In this PhD, this step has not been attempted, as similar work has already done this with historic building damage modelling (Valero, Forster, et al., 2018). Based on the position of visible defects on the masonry surface of arch bridges, it is possible to infer the underlying damage mechanism. It is envisioned that this defect location information can ultimately be extracted from the generated BIM model to automatically suggest the problems inflicting a bridge and their possible cures. For this PhD, the logic behind such an automation has been investigated, also identifying the visible defects that a classifier will need to be trained to recognise. This logic is presented below.

3.1 Disease symptom analysis for masonry arch bridges

In this thesis, the impact of different defect locations and types on the function of arch bridges, has been investigated in a way analogous to medically investigating a disease. This is presented in Figure 3-2, in which the defects that are visible on the bridge structure are labelled as symptoms. These are the cues that are used to diagnose the underlying problem with the bridge. As in the medical field, a combination of several different visual symptoms on the bridge will together point to the potential underlying problem with the bridge. Next the cures for the underlying problems are listed. These are the works that

can be done to cure the bridge of its underlying problem, which is causing the visible symptoms. Finally, there are the alleviations. These are the works that can be done to remove the visible symptoms from the bridge, without curing it of its underlying problem. In this way the visible defect on the bridge is linked to both its underlying problem and its solution. A future automated asset management tool for masonry arch bridges could therefore use the mapped information to determine the underlying problem in a bridge, and its required remedial treatment based on the detected defects and their location on a BIM model.

Figure 3-2 was mostly created by combining information from the Network Rail standards for the examination of structures (Network Rail, 2017), and the Construction Industry Research and Information Association (CIRIA) documentation for assessing the condition of and remedying masonry arch bridges (McKibbins et al., 2006). In general, the Network rail standards have been used to determine the visible symptoms that are currently sought during manual bridge visual inspections, and the CIRIA documentation used to ascertain the underlying problems and the solutions for each symptom. Additional sources that have been used to a lesser extent are Atkins (2009), and Helmerich et al. (2007), which also consider European and non-railway masonry arch bridge preservation practices. The novelty of this work is therefore in the logical mapping of the visual defects to their remedial treatments, rather than the actual content.

The utility of this mapping is demonstrated in Figure 3-3 where a subsection of the ailments and resolutions have been extracted from Figure 3-2. Here, if the visible symptoms in a bridge are a dip in the parapet, and lateral cracks in the span of the arch, then the likely underlying problem is movement at the abutments and the cure is piling through the abutments. If, however, the symptoms also include crushing of the brickwork, then the likely underlying problem is a failure of the arch barrel, with the cure being a concrete relieving arch or concrete backfill. In this way, through the position and combination of the symptoms present on the bridge, it is possible to diagnose the underlying problems and cures, in a manner that could in the future be automated.

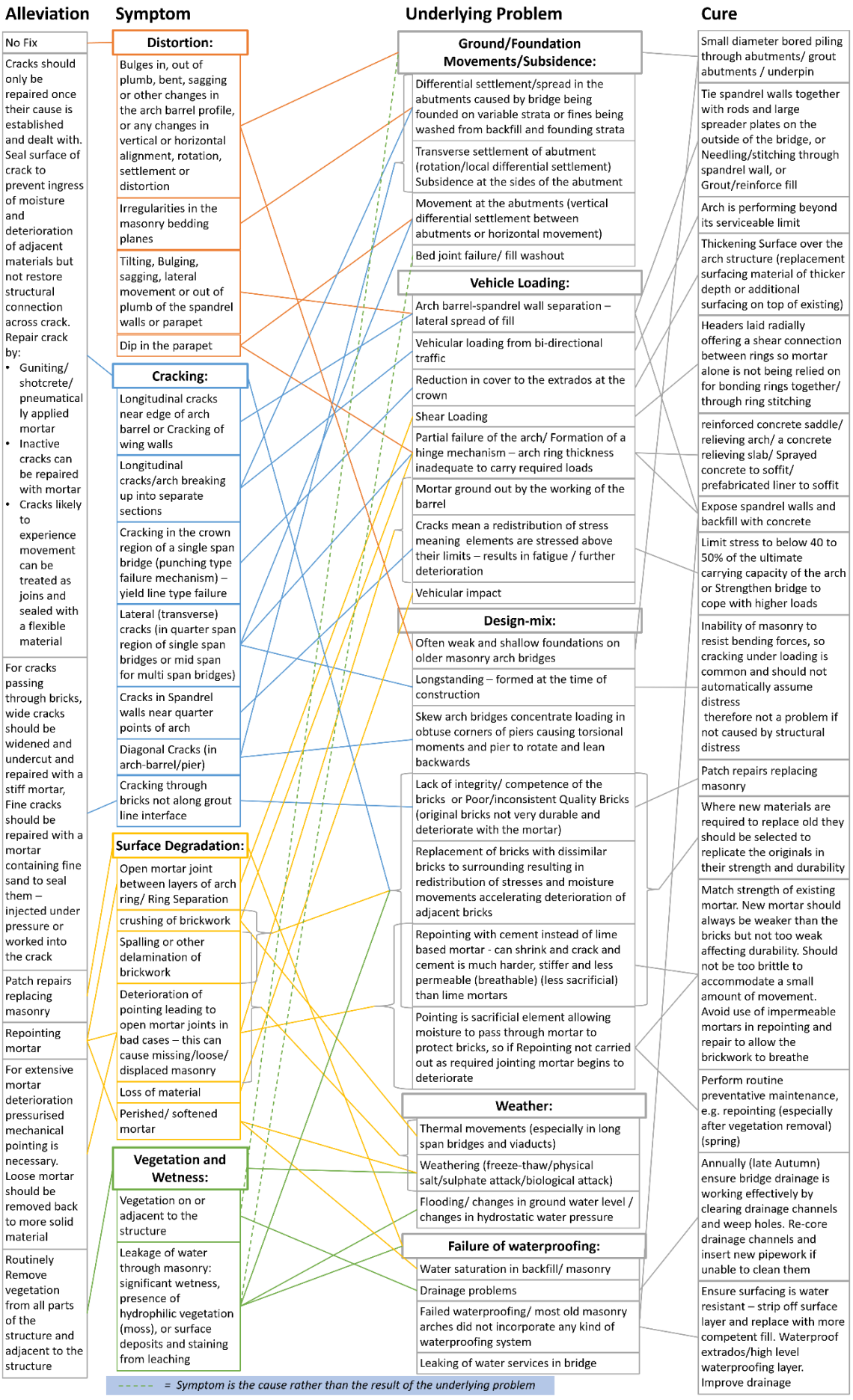


Figure 3-2: Masonry arch bridge ailments and resolutions

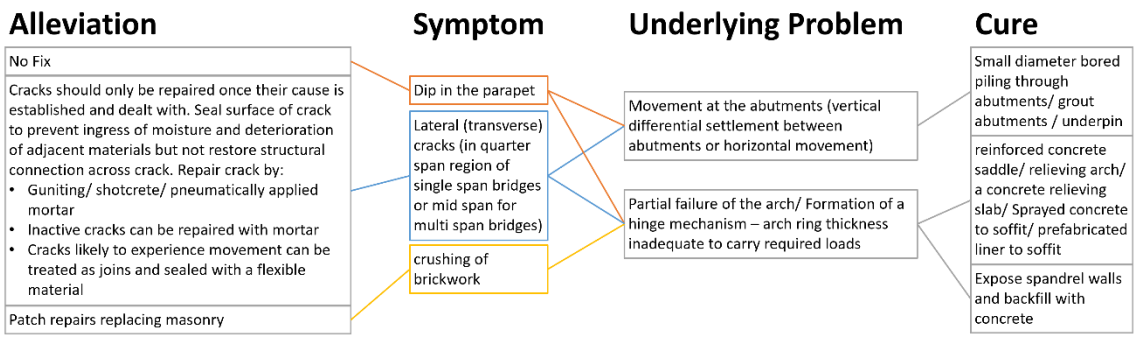


Figure 3-3: Focused subsection of bridge ailments and resolutions mapping

3.2 Important defect classes for detection

A key output to developing the disease mapping in Figure 3-2 is the determination of the defect classes that are important for detection in masonry arch bridges. It is apparent that the main classes of visible defect giving symptoms of bridge ailments are distortions in the shape of the bridge, irregularities in the mortar joints, cracking, spalling and other delamination of brickwork, missing or displaced masonry, mortar loss, vegetation, wetness, and surface deposits.

To determine the severity of each of these classes of defect on the condition of the bridge, correlation with possible Bridge Condition Marking Index (BCMI) scores from the Network Rail standards for the examination of structures (Network Rail, 2017) has been examined. BCMI scores are consistently assigned during inspections across the Network Rail bridge stock to record the deterioration of bridges from their perfect, as built condition, and assess the level of risk associated with each bridge. Scores are derived through recording defects on the elements of each bridge. For an element with no defects a severity code of A would be given. Elements are given severity ratings based on the type of defects present, their position, width, depth, length, and orientation. Non-crack defects are associated with scores ranging from B to F, and crack defects scores from G to P. Separate extent scores are also given for each defect, with the overall BCMI score for the bridge calculated from the severity and extent ratings of the defects present. BCMI ranges for each of the defect features identified have been recorded in Table 3-1 alongside an assumed severity rating (from 1 = mild to 5 = severe) based on the BCMI severity score. It is assumed that distortions in the shape of the bridge (BCMI score range of E - F) would be more accurately detected through examining the bridge geometry in its point cloud (which is outside the scope of this PhD) than by visually examining the surface, and therefore this defect class has not been considered. Although the same is also true for missing masonry where its geometry is much more prominent than its appearance, this

visually resembles mortar loss or brickwork delamination, depending on the extent of missing masonry, and therefore has been considered.

In Table 3-1 it is apparent that the least significant defect classes are wetness and surface deposits and staining, a sign of wetness. This is due to these defect classes being indicators of future problems with the bridge, rather than existing problems. Additionally, in the literature it has been suggested infrared thermography rather than visual imagery is more able to capture masonry wetness (Orbán and Gutermann, 2009). As a result, this PhD focuses on the visual detection of cracking, spalling, mortar loss, and vegetation.

Table 3-1: Severity of visible masonry defect classes

Feature	Criteria & Cause for Concern	Severity [BCMI]
Irregular mortar joints	Irregularities in the bedding planes: Sign of differential settlement of the bridge where cracks could be masked by repointing.	4 [F]
Cracking	Position, inclination, orientation, length, width, and whether it has been repointed in the past. Whether it is through bricks or along mortar joint: Cracks in certain places indicate bridge is not safe to carry load, depending on the extent and position of cracks. Longitudinal cracks within the middle third of the arch suggest the bridge is breaking up into sections. Transverse cracks within the middle third of the span suggest a mechanism is forming which is potentially very severe. Diagonal cracks suggest the bridge is in a dangerous state. Hairline cracks in individual bricks may be the result of inherent material properties.	5 [F, G-P]

Spalling	Depth of spalled / delaminated and substantially weakened material relative to original face of bridge, proportion of surface affected: Fabric of the bridge is deteriorating, suggesting work is required to repair.	3 [B-D]
Loose and missing masonry	Presence of loose or displaced masonry, proportion of surface affected: Suggests that maintenance has not been done and mortar bond has failed and/or bricks forced out of position. There is a potential for the hazard of falling bricks	4 [Ex –F]
Mortar loss	Maximum and typical depth lost. Proportion of joints showing signs of deterioration: Loosens masonry units, reducing ability of masonry to transmit and evenly distribute forces. Fabric of the bridge is deteriorating, suggesting work is required to repair.	3 [C]
Vegetation	Number, size and girth of trees and extent of bushes growing on the structure. Presence of large, or hydrophilic vegetation (moss): Tree roots and other undealt with vegetation can cause severe damage to the structural fabric of the bridge. Significant vegetation could also displace masonry and can hide other defects. Smaller organisms that may be found in damp areas of the bridge fabric can cause deterioration by increasing porosity and facilitating leaching and other mechanisms.	2 [B]

Wetness	Evidence of the presence or effect of water (excluding exposure to rain) in sufficient quantities, and proportion of structure affected: Increases the vulnerability of masonry to environmental agents causing and accelerating deterioration. Movement of moisture can result in washout of fines from particulate materials (e.g., fill) causing weakening and instability.	1 [B]
Surface deposits and staining	Proportion of surface affected: A sign of wetness, and the washout of the vulnerable calcium hydroxide and carbonate components of mortars. Their loss creates secondary porosity that can weaken materials and in turn aggravates the effect of other agents like freeze-thaw.	1 [B]

3.3 Chapter summary

In this chapter a workflow for automatically determining underlying problems with masonry arch bridges has been proposed. This workflow uses geometry and image data to detect defects visible on the surface of the bridge, and to locate these defects in a visual 3D BIM model.

The focus of this chapter has been devising a mapping between the underlying problems facing masonry arch bridges and their visible defect symptoms, akin to a medical diagnosis, with the potential for future automation. This process has also been used to identify the major visible defects that an automated classifier should be trained to detect.

4 DATASET GENERATION

Previous research on the detection of defects in images has highlighted how fundamental the training dataset is to successful classification (Shin et al., 2016). It has been suggested that factors important to a quality dataset are the size of the dataset, how similar the dataset is to real world conditions and the variation of both environmental and surface conditions captured in the dataset. Limited research has been focused on creating a dataset of realistic masonry images, i.e., images that contain the real variability of masonry condition observed on real masonry arch bridges. Therefore, the creation of a thorough dataset of masonry arch bridge surfaces is a major part of the work for this PhD.

4.1 Equipment used

For each site visited, both image and geometry data have been captured, using a digital camera and laser scanner, respectively. Based on equipment availability, different sensors were used for different sites, with the following equipment used across the different sites:

- Cameras:
    - Cannon EOS 7D Mark II: 20MP APS-C sensor
    - Sony Alpha 7R II: 42.2MP full-frame Exmor R CMOS sensor
    - JVC GC-PX10: 12MP 1/2.3” back-side illuminated CMOS sensor
- Laser Scanners:
    - Faro Focus 3D X330: 0.6-330m range. ±2mm ranging error at 25m range
    - Faro Focus 3D S120: 0.6-120m range. ±2mm ranging error at 25m range

The camera was used mounted on a tripod to reduce image blur and used at the same time as the laser scanner was running. Photos have been taken in a way to ensure sufficient

resolution of the masonry surface for detail to be captured, and such that there is sufficient overlap between image frames to enable image stitching. Additionally, photos at various levels of detail have been taken to aid image stitching and Structure from Motion (SFM) photographic model reconstruction.

Laser scanning positions have been chosen around the captured bridges to prevent occlusions, such that the whole surface of the bridge is measured. Point clouds from different scan positions have been registered to produce a single point cloud for each bridge, using the Faro Scene software (Faro, n.d.). Registration was done by means of both reference markers placed while scanning, and cloud-based alignment. Aligned point clouds have been produced for all the bridges visited which are listed in Chapter 4.2. An example of a 2D view of one of the colourised point clouds produced is shown in Figure 4-1, depicting Hertford Viaduct.

Figure 4-1: Point cloud for Hertford Viaduct

4.2 Sites visited

Data has been collected from several masonry viaducts and bridges, the majority nearby to Cambridge. The capture of many different bridges means that the dataset generated depicts a wide range of masonry conditions, making it more suited for training a classifier to generalise over multiple structures. The bridges captured in the dataset have all been constructed from brick masonry. In total, data was collected from 9 viaducts and 4 bridges. Across the viaducts a total of 40 spans were captured, meaning that an overall total of 44 spans have been captured, constituting a significant total surface area of

captured masonry of 15,308m^2 in total (surface area manually measured from point cloud for each bridge captured). Across these spans, the dataset created consists of 137 individual laser scans and 28,481 photos. A breakdown of the bridges visited is shown in Figure 4-2, with additional information on each bridge detailed following. For clarity, the engineer's line and bridge references are listed in brackets for each bridge. The dates on which data was collected are given in the dd/mm/yy format.

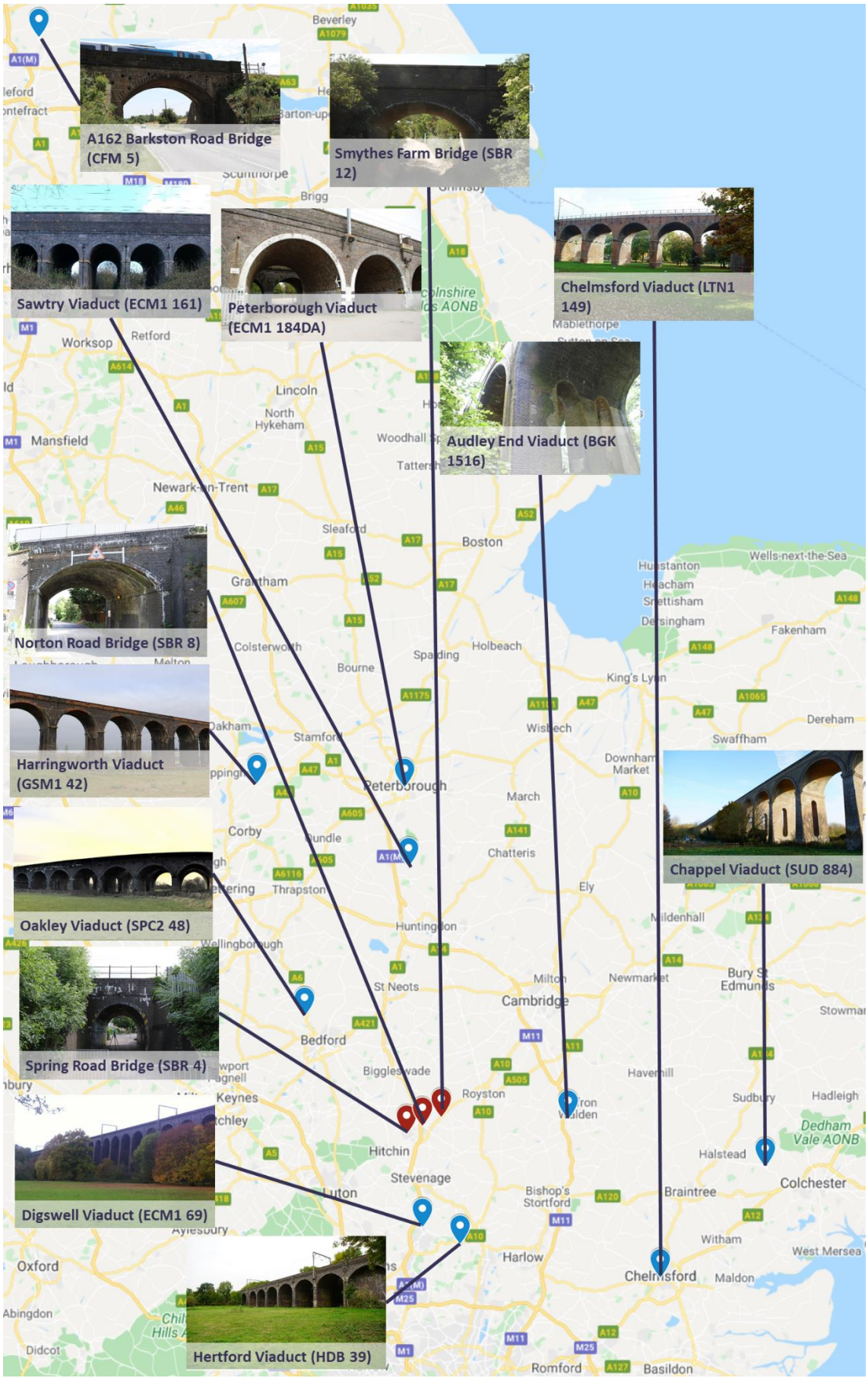


Figure 4-2: Map showing sites visited for data collection

- **Audley End Viaduct (BGK 1516)**
    - Spanning a steep valley just south of Audley End railway station in Cambridgeshire, 18m high at its centre. It consists of 7 arches, each with a 6m span. Constructed in 1843, it carries the West Anglia Main Line running between London Liverpool Street and Cambridge (Audley End Viaduct n.d.).
    - Data collected on 10/08/17 using the Cannon camera and S120 scanner
    - 2 Spans captured, with total captured surface area approximately 525m^2
    - 1198 images and 6 laser scans taken

    - Key features and distractors: Overall good condition masonry with well-defined and clean mortar joints. Some areas of graffiti and block sections of paint as well as vegetation obscuring mortar joints. Photos taken with neutral lighting and minimal shadowing.

- **Chappel Viaduct (SUD 884)**
    - Spanning 320m, 23m above the valley of the river Colne, south of Chappel & Wakes Colne railway station in Essex. It is grade II listed, and consists of 32 arches each with a span of 9.1m. Constructed in 1849, it carries the Gainsborough line, a branch line off the Great Eastern Mainline (Historic England, 1967).
    - Data collected on 26/11/16 using the Sony camera and X330 scanner
    - 4 Spans captured, with total captured surface area approximately 1788m^2
    - 862 images and 15 laser scans taken

    - Key features and distractors: Overall good condition masonry with clean and shallow mortar joints, though brick face is rough and has spalled in parts. Minimal vegetation, though heavy shadowing in photos. Areas of wetness and efflorescence on arch barrels.

- **Chelmsford Viaduct (LTN1 149)**
    - Chelmsford Viaduct carries the Great Eastern Mainline 16.5m above Central Park and the River Can, just south of Chelmsford Station in Essex. It consists of 18 arches, each spanning 9.5m and was constructed in 1842 (River Can and Occupation Road Viaduct n.d.).
    - Data collected on 31/10/16 using the Sony camera and X330 scanner.
    - 1 Span captured, with total captured surface area approximately 351m^2
    - 424 images and 15 laser scans taken

    - Key features and distractors: Mostly good condition masonry with several areas having undergone repointing or replacing of bricks. A few areas have graffiti, or block paint to cover graffiti, and vegetation is present in some areas. Some photos show shadowing on the brickwork.

- **Digswell Viaduct (ECM1 69)**
    - A grade II* listed viaduct which was opened in 1850 and consists of 40 arches each with a span of 9m. It carries the East Coast Mainline for 475m, 30m above the valley of the river Mimram. It is south of Welwyn North railway station in Hertfordshire (Historic England, 1980).
    - Data collected on 14/11/16 with Sony camera and X330 scanner for 2 spans and on 05/11/17 with Canon camera and X330 scanner for 4 spans.
    - 6 Spans captured, with total captured surface area approximately 4074m^2
    - 5815 images and 22 laser scans taken

    - Key features and distractors: Excellent condition masonry with well-defined and clean mortar joints and smooth brick surfaces. Almost no vegetation, though a few areas of efflorescence on arch barrels. Photos taken with neutral lighting and minimal shadowing.

- **Harringworth Viaduct (GSM1 42)**
    - The Harringworth viaduct is the longest masonry viaduct across a valley in the UK, spanning 1.166km, 18m above the valley of the River Welland between Harringworth in Northamptonshire and Seaton in Rutland. It consists of 82 arches, each with a 12m span. Completed in 1878, and grade II listed, it carries the Oakham to Kettering line (Historic England, 1987).
    - Data collected on 17/01/17 using the Canon camera and X330 scanner
    - 4 Spans captured, with total captured surface area approximately 1920m^2
    - 1773 images and 13 laser scans taken

    - Key features and distractors: Masonry in mixed condition with areas of wetness and surface deposits, though minimal vegetation. Some mortar recesses are deep and on the arch barrel mortar joints are less well defined. Photos taken with neutral lighting and minimal shadowing.

- **Hertford Viaduct (HDB 39)**
    - Carries the Hertford loop line, a branch and diversionary route of the East Coast Main Line for 280m, 13.5m above the river Mimram to the south of Hertford North Station in Hertfordshire. It consists of 14 arches, each spanning 10m and was constructed in 1915 (Hertford Viaduct n.d.).
    - Data collected on 29/10/16 using Sony camera and X330 scanner for 1 span and on 06/11/17 using Canon camera and X330 scanner for 5 spans.
    - 6 Spans captured, with total captured surface area approximately 2070m^2
    - 6082 images and 26 laser scans taken

    - Key features and distractors: Mostly good condition masonry with smooth brick faces and well-defined mortar joints. Some vegetation and surface deposits present. Photos mostly taken with neutral lighting, though some experience shadowing.

- **Oakley Viaduct (SPC2 48)**
  - 2 parallel viaducts, each carrying 2 tracks of the Midlands Mainline for 120m, 9m above the River Great Ouse. Each viaduct has 11 arches, each spanning 8m. It opened 1857 and is just north of the village of Oakley in Bedfordshire (Oakley Viaduct n.d.).
  - Data collected on 16/02/18 using the Canon camera and X330 scanner
  - 12 Spans captured, with total captured surface area approximately 2724m^2
  - 3154 images and 17 laser scans taken

  - Key features and distractors: Masonry in average condition with mostly smooth brick faces, though in some areas spalling is prevalent. Some mortar joints are very deep, whilst staining and wetness distracts from others. Some graffiti is present on the structure. Photos mostly taken with neutral lighting, though some experience shadowing.

- **Peterborough Viaduct (ECM1 184DA)**
    - 2 parallel viaducts carrying the East Coast Mainline 8m above the ground just south of Peterborough railway station. The first viaduct was built in 1850 and the other in 1924 when the line was quadrupled. The first viaduct spans 62m, and has 8 arches, each spanning 6m. The second spans 76m, and has 10 arches, each spanning 6.5m (Peterborough Viaduct n.d.).
    - Data collected on 27/11/16 using the Sony camera and X330 scanner
    - 4 Spans captured, with total captured surface area approximately 656m^2
    - 1876 images and 9 laser scans taken

    - Key features and distractors: Masonry in average condition, though fully painted on piers. Arch barrel masonry shows surface deposits with little contrast between mortar and brick. Photos mostly taken with neutral lighting, though some experience shadowing.

- **Sawtry Viaduct (ECM1 161)**
    - A viaduct spanning 25m with a height of 9.5m over Red Drove track to the east of the village of Sawtry in Cambridgeshire. It has 5 arches, each spanning 3.7m. Carrying the East Coast mainline, it was originally built in 1850, but was widened in 1924 when the line was quadrupled (Ecm1/161 Viaduct n.d.).
    - Data collected on 21/03/17 using the JVC camera and X330 scanner
    - 1 Span captured, with total captured surface area approximately 193m^2
    - 372 images and 2 laser scans taken

    - Key features and distractors: Masonry in average condition, with mostly clear well pointed mortar joints and smooth brick faces. Some areas show spalling and other areas surface deposits. There is minimal vegetation and graffiti. Photos taken with neutral lighting and minimal shadowing.

- **A162 Barkston Road Bridge (CFM 5)**
    - A skew arch bridge opened in 1869 over the A162 south of the village of Barkston Ash in North Yorkshire. It has a height of 5.6m and a span of 10.6m. It carries the direct railway traffic between Leeds and York railway stations (A162 Barkston Road Underline Bridge n.d.).
    - Data collected on 05/07/18 using the Canon camera and X330 scanner
    - Total captured surface area approximately 118m^2 across bridge span
    - 355 images and 5 laser scans taken

    - Key features and distractors: Only the arch barrel contains brick masonry. In general brick masonry in good condition with clean mortar joints, and smooth brick faces, though towards the springing line there is the presence of leaching and deposits. Photos taken with neutral lighting and minimal shadowing.

- **Spring Road Bridge (SBR 4)**
    - An underline bridge on the Cambridge line, connecting Cambridge to the East Coast Mainline which opened in 1851. It has a height of 3.5m and a span of 3.66m (Sbr/4 Spring Rd Underline Bridge n.d.).
    - Data collected on 06/06/19 using the Canon camera and X330 scanner
    - Total captured surface area approximately 196m^2 across bridge span
    - 2085 images and 2 laser scans taken

    - Key features and distractors: Masonry generally in good condition with smooth brick surfaces and well-defined mortar joints. However, most of the surface is painted, either with graffiti or block painting. Other exposed areas suffer from efflorescence and surface deposits. Internal photos taken with neutral lighting, but most others show shadowing, especially on the mortar interface.

- **Norton Road Bridge (SBR 8)**
    - A skew arch bridge on the Cambridge line which opened in 1851. It has a height of 4.92m and a span of 6.11m (Sbr/8 Norton Road Underline Bridge n.d.).
    - Data collected on 06/06/19 using the Canon camera and X330 scanner
    - Total captured surface area approximately 331m^2 across bridge span
    - 2442 images and 2 laser scans taken

    - Key features and distractors: Masonry generally in poor condition with rough brick surfaces and unclear mortar. Most of the masonry surface is covered in soot and other deposits. In some areas mortar joints are very deep. Photos mostly taken with neutral lighting, though a few experience shadowing, especially around mortar joints.

- **Smythes Farm Bridge (SBR 12)**
    - An underline bridge on the Cambridge line which opened in 1851. It has a height of 4m and a span of 6.1m (Sbr /12 Underline Bridge n.d.).
    - Data collected on 06/06/19 using the Canon camera and X330 scanner
    - Total captured surface area approximately 362m^2 across bridge span
    - 2043 images and 3 laser scans taken

    - Key features and distractors: Masonry generally in good condition with smooth brick surfaces, though the abutments have a covering of mud from farm traffic. Areas of the arch barrel also show significant efflorescence and deposits. Photos were taken with bright sunlight, so shadowing is prevalent, especially around mortar joints.

4.3 Image resolution specification

To ensure a sufficient level of detail is captured in photographs taken of bridge surfaces, photographs have been captured with the masonry surface imaged at a resolution greater than a predetermined minimum. This has been determined based on the size of the smallest defect that is necessary to be visible in photographs, namely a 1mm thick crack. Thinner cracks are likely to be those isolated to a single brick. These are likely to be a result of material properties, and so are less important. The Nyquist Sampling Theorem states that to reproduce a signal, it should be sampled at twice the highest frequency components. In images the pixel size should therefore be ½ the size of the smallest object (Ruzin, 2011). For recording 1mm cracks, an image surface resolution of 2 pixels per mm in each dimension is therefore required. McRobbie (2015) has suggested that dark, sub-pixel features will affect the light detected by each pixel, reducing the pixel value, making sub-pixel features visible. This suggests that with surface resolutions of lower than 1 pixel per mm, 1mm thick cracks will still be visible. However, images have been captured at a resolution of 2 pixels per mm, as this will enable crack width measurements down to 1mm to be made. The resolution of the captured surface texture is governed by the equation:

$$texture\ resolution\ (px/mm) = \frac{camera\ resolution\ (px)}{field\ of\ view\ (mm)} \qquad (4\text{-}1)$$

The resolution of the captured surface texture is therefore governed by the field of view of the image, i.e., the surface area of masonry in the image, and the resolution of the camera. For a given camera the resolution is fixed, so the texture resolution can only be changed by changing the field of view. Therefore, to ensure a texture of at least 2 pixels per mm, different maximum field of views were specified for the three cameras used to capture data.

- The Canon camera has a 20MP sensor, with resolution 5472 x 3648 pixels giving a maximum field of view of 2.7 x 1.8m of surface.
- The Sony camera has a 42.2MP sensor, with resolution 7952 x 5304 pixels, giving a maximum field of view of 4.0 x 2.7m.
- The JVC camera has a 12MP sensor, with resolution 4000 x 3000 pixels giving a maximum field of view of 2.0 x 1.5m.

The field of view is governed by the equation:

$$field\ of\ view\ (mm) = \frac{sensor\ size\ (mm) \times working\ distance\ (mm)}{focal\ length\ (mm)} \qquad (4\text{-}2)$$

The size of the sensor is a property of the camera and is fixed, so changing the field of view is achieved through changing the focal length of the lens, or by changing the working distance, i.e., the distance between the camera and the surface. While collecting field data, a telephoto lens was used, allowing a range of focal lengths, so both this and the imaging position were changed to ensure the correct field of view.

In addition to the high detail images, images of the surface at lower resolutions, and taken from non-orthographic angles have also been taken to aid with image stitching, and structure from motion model reconstruction.

4.4 Image stitching and orthophoto generation

Stitching of individual images for generation of orthophotos depicting complete bridge elements has been investigated by McRobbie (2015) who has evaluated different image stitching solutions for highway bridges, concluding that relying on image features for image stitching can go wrong when there are either too many or not enough unique image features. They have proposed generation of orthophotos of bridges, through stitching images using an automated pan-tilt camera mount.

For this work, stitching using image features has been used, as it was found there were sufficient features in a masonry surface for alignment of images. Stitching has been attempted using both Microsoft Image Composite Editor (Microsoft, 2017) and Hugin Panorama Photo Stitcher (OpenSource, 2020). It was found that Hugin gave much superior results due to the greater level of control that was afforded to the process. Stitched panoramas have been re-projected orthogonal to the masonry surface to generate orthophotos, meaning there is no perspective distortion. Panoramic orthophotos have been constructed for a pier of a single span of three of the viaducts in the dataset:

- **Hertford Viaduct**: Stitched from 52 individual images, the resulting panorama measures 27872 x 21044 (w x h) pixels. The pier was measured to be 9.22m wide from its point cloud, meaning that the texture resolution is 3.0 pixels per mm.
- **Peterborough Viaduct**: Stitched from 38 individual images, the resulting panorama measures 43100 x 7113 (w x h) pixels. The pier was measured to be 8.34m wide in its point cloud, so the texture resolution is 5.2 pixels per mm.
- **Chelmsford Viaduct**: Stitched from 38 individual images, the resulting panorama measures 24325 x 20087 (w x h) pixels. The pier was measured to be 7.99m wide in its point cloud, meaning that the texture resolution is 3.0 pixels per mm.

For reference, the complete stitched orthophoto for a pier of Hertford Viaduct is shown in Figure 4-3. For most of the panorama, the interface between the comprising images is seamless. However, in a few places, as in Figure 4-4, there is a small degree of misalignment. However, this is not significant enough to affect the ability of a classifier to detect features on the surface. Additionally, where panoramic photo stitching is attempted on the arch barrel, this results in significant misalignment as shown in Figure 4-5. Here the panorama has been cropped to ensure it is at a resolution that is viewable, so only a quarter of the barrel is visible. The left edge of the image is aligned with the spandrel wall, and the bottom edge with the springing line. In this figure, the misalignment of the individual images is obvious due to the nonlinear bed joints. This misalignment is due to attempting to re-project images of a curved surface onto a flat plane.

Figure 4-3: Orthographic stitched panorama for Hertford Viaduct

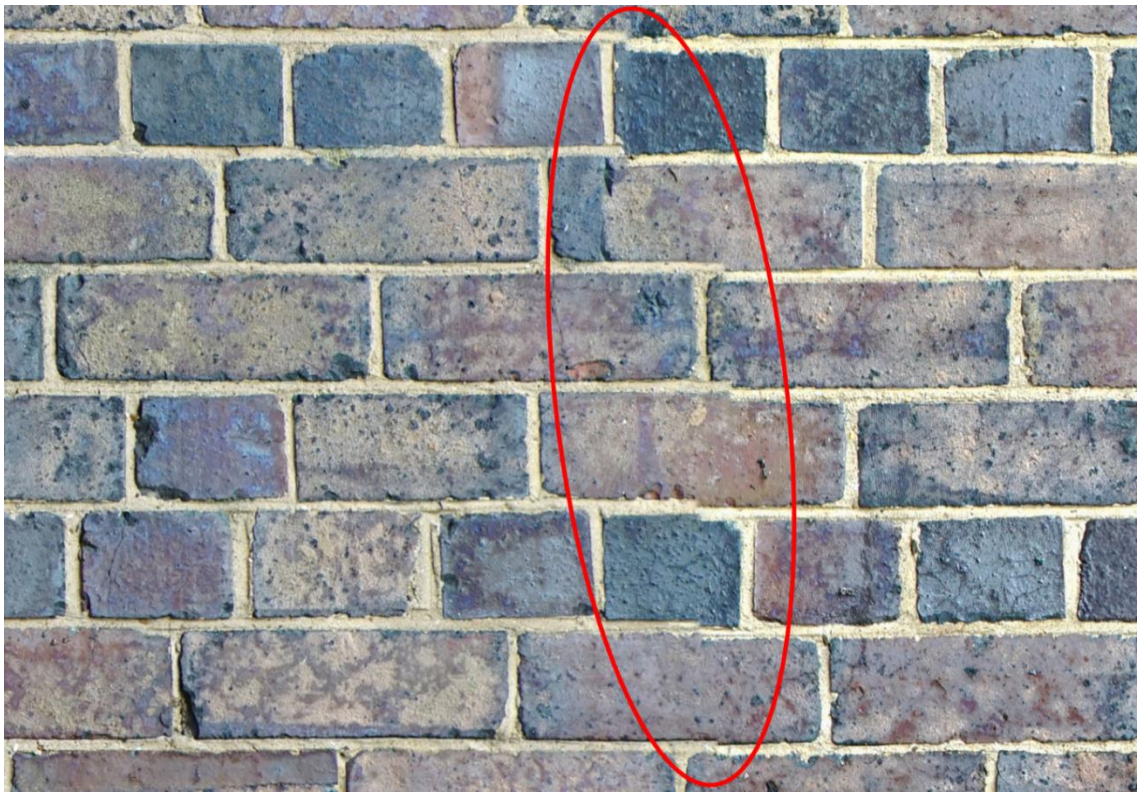


Figure 4-4: Misalignment in image stitching process

Figure 4-5: Misalignment in image stitching for Digswell Viaduct arch barrel

4.5 Photogrammetry

As a result of the misalignment shown in Figure 4-5, it can be determined that panoramic image stitching is not powerful enough to build seamless models of masonry arch bridges, due to the curvature of their arch barrels. As a result, photogrammetry has been used. This looks to recover 3D structure from projected 2D images utilising Structure from Motion (SFM) and Multi-View Stereo (MVS). This works on the principle that when an observer moves, objects around them move by different amounts depending on their distance from the observer. By matching unique points in images taken from various positions, 3D structure can then be determined, creating a 3D point cloud onto which images can be mapped. This also means that when the arch barrel texture is built, it is re-projected onto a 3D plane with curvature matching that of the barrel, such that misalignments such as in Figure 4-5 will not occur.

Many software packages exist for performing photogrammetry. For this work, several of these have been trialled to determine suitability. These are:

- Agisoft Metashape (Agisoft, 2019)
- Autodesk Recap (Autodesk, 2017)
- Colmap (Schönberger, 2020)
- AliceVision Meshroom (Griwodz et al., 2021)
- Multi-View Environment (Fuhrmann et al., 2014)
- OpenMVS (cdcseacave, 2019)
- Visual SFM (Wu, 2011)

Of these, Agisoft Metashape and Autodesk Recap are paid commercial products, whereas the others are all open source or freeware software. It was found that both the reconstruction performance, and ease of use of Agisoft Metashape was best, and as a result was chosen for developing photogrammetric models. Most of the examined software have now had new versions released since being tested, so this performance gap may not still be the case today. Other, more costly photogrammetry software exists, though these were not evaluated due to funding constraints, and the performance of Agisoft Metashape being adequate for reconstructing 3D models from the dataset given.

For 3D model reconstruction in Agisoft Metashape, the workflow is as follows. Initially individual photos are aligned using SFM to compute the camera position for each photo and generate a sparse point cloud of the 3D model. Next a dense point cloud is calculated using MVS, whereby depth maps are computed for overlapping image pairs. At this point,

it would have been possible to import the point cloud that was developed through laser scanning instead of calculating a dense point cloud through photogrammetry. However, the increased accuracy of the laser scan derived point cloud was not required, so for simplicity the photogrammetric point cloud was used. A mesh of the surface is built from the dense point cloud, through triangulation of the points. This mesh surface is then colourised based on the projected image data to form a colourised texture, and a colourised 3D model of the bridge. Photogrammetry has been performed on: Spring Road Bridge (SBR 4), Norton Road Bridge (SBR 8), and Smythes Farm Bridge (SBR 12). These bridges are located by red markers in the map in Figure 4-2. The colourised 3D photogrammetric model for Norton Road Bridge (SBR 8) is shown in Figure 4-6 for reference.

Figure 4-6: Photogrammetric model for Norton Road Bridge (SBR 8)

4.5.1 Creating 2D image textures

The work of this PhD focuses on detecting defects from the visual features of masonry arch bridges rather than geometric features. As a result, the methodology followed classifies masonry surfaces as defective or not from 2D image textures. It is therefore necessary to extract 2D image textures from the developed 3D models.

4.5.1.1 Orthophotos

As discussed in Chapter 2.2 in the literature review, the current practice for extracting 2D textures from 3D models is through capturing an orthographic projection of the texture being studied. For flat surfaces such as bridge piers, this would give a similar output to that from an orthographic projection of stitched photos as in Figure 4-3. However, an

orthophoto of a curved surface such as an arch barrel would result in non-uniform scaling across the image as in the Network Rail data in Figure 2-4 in Chapter 2.2, and the data collected for this work presented in Figure 4-9.

4.5.1.2 UV mapping

UV maps describe the mapping of 2D image textures onto a colourised 3D model, where the 2D textures are positioned in the “U”, “V” coordinate system, which is mapped to the “X”, “Y”, “Z” coordinate system of the 3D model. When colourising a mesh in Agisoft Metashape, the different possible mapping modes are an orthophoto mapping mode and a generic mapping mode. As previously discussed, an orthophoto mapping of a curved surface such as an arch barrel would lead to scaling. The generic mapping mode automatically divides the 2D texture map arbitrarily to attempt to create as uniform a texture as possible. Although this results in a texture with minimal scaling, this is not useful as the 2D texture in the UV map is discontinuous. An example of a UV map generated with a generic mapping is shown in Figure 4-7.

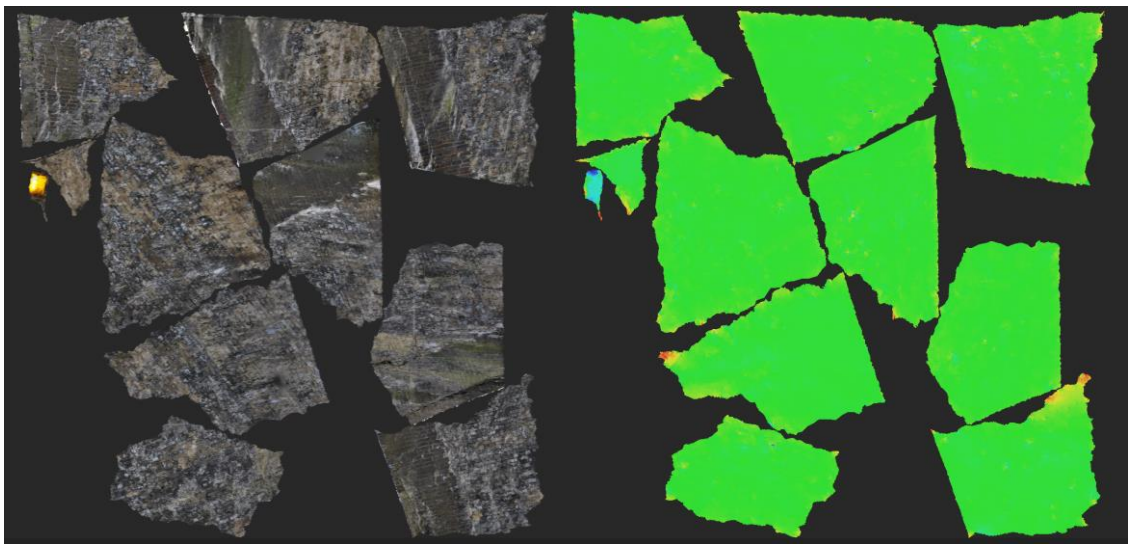


Figure 4-7: UV Map for the barrel of Norton Road Bridge (SBR 8) with generic mapping mode. Left image shows texture colour, and right image shows scaling

Arch barrels on masonry arch bridges are only single curved surfaces, so through unravelling the curved surface, it is possible to create a UV map 2D texture with no scaling or cuts, in a way that would not be possible for a double curved surface. To do this the UV map has been defined outside of Agisoft Metashape, and then reimported back in for colouration. This has been done by UV unwrapping in Blender (Blender Online Community, 2018), an open source 3D creation suite.

2D textures from unwrapped UV maps have been created separately for each of the distinct parts of a bridge (e.g., arch barrel, spandrels, wing-walls, etc.) to simplify the geometry of the unwrapping process, such that the UV for each part is undivided with minimal scaling. An example 2D texture for the arch barrel and abutments of a bridge is shown in Figure 4-8. Notice that unlike in Figure 4-5 there is no scaling or discontinuities between the individual photos.

UV unwrapping the arch barrel as well as orthophoto mapping are compared in Figure 4-9. From the scaling mapping, it is apparent that there is much more scaling in the orthophoto mapping Figure 4-9d-i compared to the unwrapping mapping Figure 4-9c-i. The colour scales of the two figures are independent based on the scale range within the figure individually. The scale range for the unwrapping mapping Figure 4-9c-i is from a minimum of 85.9% to a maximum of 115%. This range is caused by imperfections in the shape of the arch such as a light fitting. In contrast the scale range for the orthophoto mapping Figure 4-9c-i is from a minimum of 19.3% to a maximum of 116%. There is therefore a much larger scale range for orthophoto mapping, as well as much larger areas of the texture being scaled as shown in the figure.

Figure 4-9 in c-ii / c-iii and d-ii / d-iii compares the image textures for unwrapping and orthophoto based UV mapping, respectively. A window of the same masonry image texture from the crown of the arch and near the springers has been captured for both mappings in Figure 4-9 in, ii and iii, respectively. The mappings have been scaled such that the texture near the crown is the same scale. As can be seen, the texture near the springing for the orthophoto mapping is scaled compared to both its texture near the crown and the springing texture for the UV mapping. This is a similar result to what is visible in the Network Rail orthophoto mapped data of an arch barrel discussed in the literature review and shown in Figure 2-4.

As a result of this scaling, all 2D image textures used in this work have been generated through UV unwrapping, and it is proposed that this is the method that is adopted generally for generating 2D image textures of arch bridge barrels.

Figure 4-8: 2D texture for unwrapped UV from textured 3D geometric model of the arch barrel and abutments of Smythes Farm Bridge (SBR 12)

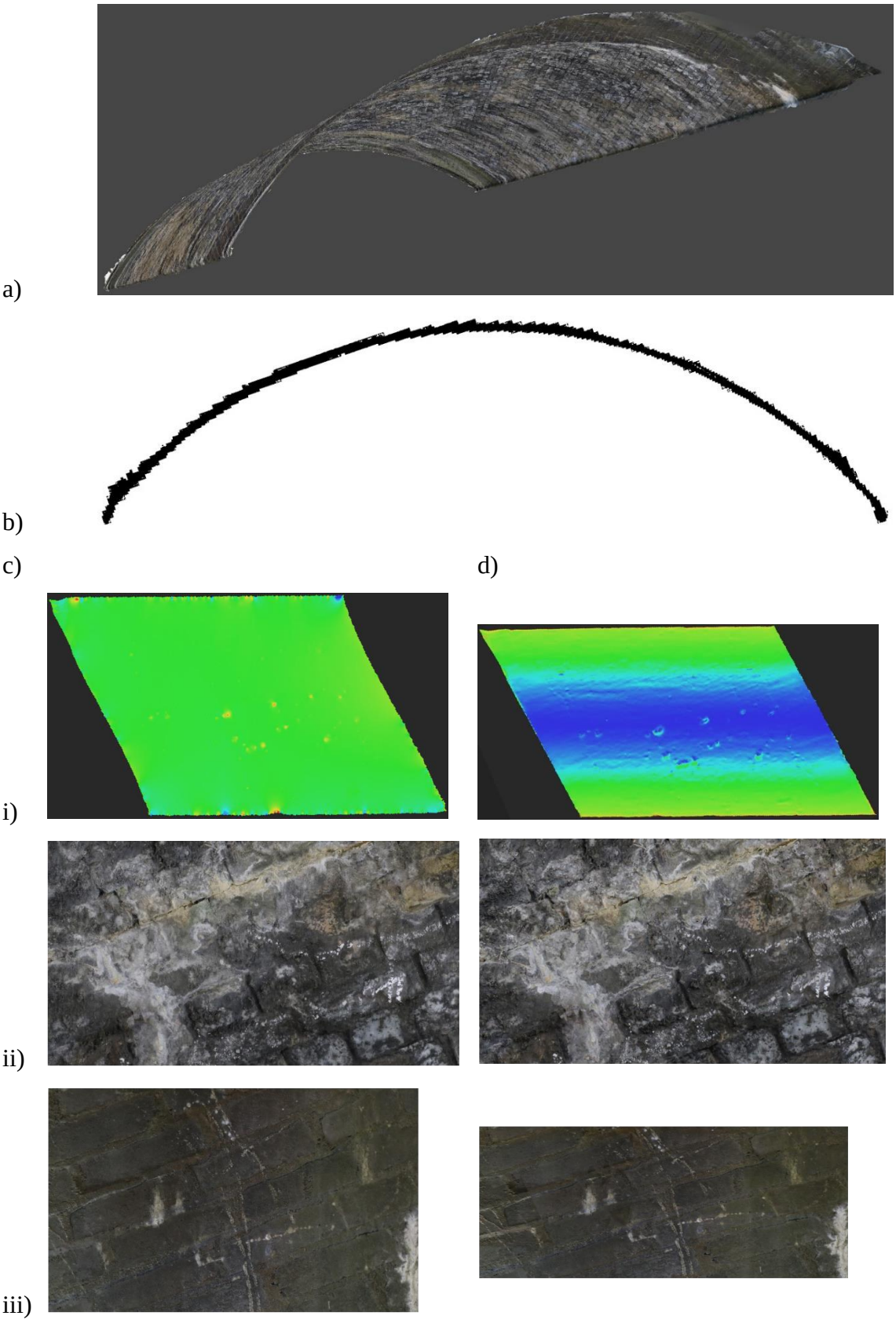


Figure 4-9: UV Map scaling effect on arch barrel of Norton Road bridge (SBR 8): a) Barrel model, b) Barrel profile, c) UV unwrap mapping, d) Orthothophoto UV mapping; i) barrel scaling, ii) texture at barrel crown, iii) texture close to barrel springer

4.6 Texture annotation

A key part of the work in creating a dataset of masonry arch bridge surface image data is the annotation of that data. Annotated data is needed for both the training and testing of classification algorithms. Image textures of masonry surfaces with an orthogonal projection to the surface (no perspective distortion) are the expected input of a defect detection algorithm. This is because it is envisioned that defect detection will be performed on image-textured three-dimensional models of masonry arch bridges as part of an automated bridge inspection process. These orthographically projected textures have been generated in two separate ways as a part of two separate datasets.

The first dataset uses image data from those bridges with blue markers on the map in Figure 4-2 (predominantly viaduct images). From all the image data of these bridges, 94 images have been chosen based on their depiction of defects. This is done to try and maximise the prevalence of defects in the dataset, as most of the masonry surface area does not contain defective features. On this dataset the images have had their perspective distortion corrected to form orthogonally projected textures by re-projecting images individually using the Hugin Panorama Photo Stitcher software. This was done through marking horizontal and vertical alignments (bed and head mortar joints) depicted in the photos and forcing their alignments in the re-projected texture. Additionally, the three orthographic panoramic images that were presented in Chapter 4.4, which depict complete piers of 3 of the viaducts captured, are included in this dataset.

The second dataset has been generated from photogrammetric models of bridges as described in Chapter 4.5.1.2. The bridges in this dataset are shown with red markers on the map in Figure 4-2. The whole masonry surface of these bridges has been included in this dataset (approximately 890m^2), so the prevalence of defects in the dataset is the same as that in the bridge itself.

For both datasets, the image textures have been resized to ensure a constant surface resolution in all images, as this would be achievable automatically with 2D image texture data generated from geometric 3D models. This resolution has been determined by standardising the average number of pixels for a brick course in each image to 155 which is approximately equal to a resolution of 2 mm per pixel, the resolution defined in Chapter 4.3.

All the generated image textures have been annotated pixel-wise with the different defect classes. This has been done by manually annotating the pixels in the images where a

defect is present. The defect classes that were annotated are outlined in Chapter 3.2. An example of this for one of the annotated images is shown in Figure 4-10. As well as annotating defects, the pixels containing mortar joints have also been annotated to consider automatically detecting mortar joints and to determine the effect of mortar joints on defect detection performance. Annotation has been done using a pen-input touchscreen display, with the Fire Alpaca digital painting software (FireAlpaca, 2021). Each defect class has been annotated within its own image layer, with the base masonry texture also in its own image layer. These image layers have then been exported as separate, interrelated images.

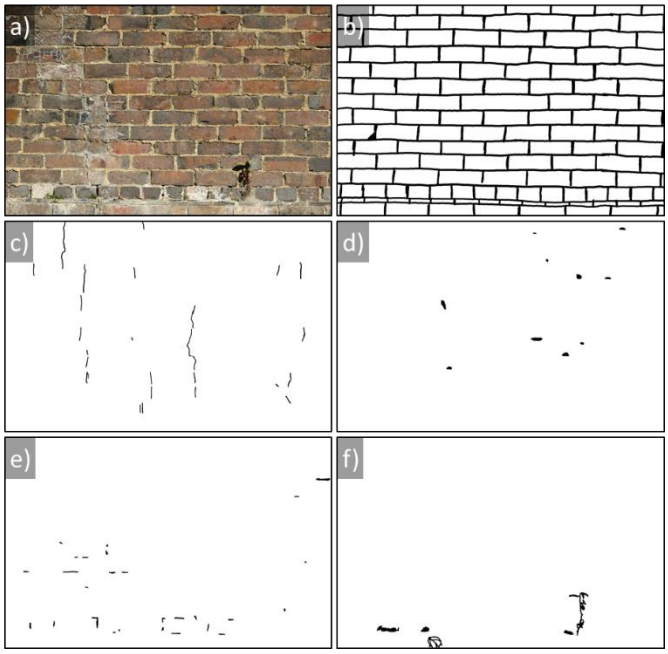


Figure 4-10: Image annotations for recording defect locations showing: a) original image, b) mortar joints, c) crack locations, d) spalling locations, e) mortar loss locations, f) vegetation locations

4.7 Chapter summary

This chapter has outlined the generation of the dataset that has been created as part of this PhD. This has formed a significant part of this PhD, as a well-defined dataset is viewed as essential to creating a successful classifier. The dataset that has been created encompasses a large amount of masonry data, the most comprehensive dataset by a considerable margin in comparison to other works that have been examined in the literature.

Additionally in this chapter, a workflow has been established for creating 2D image textures of masonry arch bridges. In terms of infrastructure assets, masonry arch bridges are unique in having a non-flat surface that results in scaling when viewed with an orthophoto projection. The proposed workflow looks to generate UV maps from image-textured geometric models through using UV unwrapping, to generate non-scaled, non-segmented, masonry surface image textures.

Finally, the process of annotating the masonry dataset has been discussed. This annotated dataset forms the backbone of the work discussed in the remainder of this thesis.

5 MORTAR JOINT DETECTION

In the literature review it was identified that detection of mortar joints in masonry surfaces can be beneficial because:

- Mortar joint location could aid defect detection through simplifying heterogeneous masonry surface,
- Mortar joint distortions can be useful for highlighting historic settlement and deformations of masonry arch bridges,
- Mortar joint locations help quantify sized of damaged areas for estimating material requirements for repair,
- Mortar joint locations can improve the accuracy of modelling bridges for calculation of their service capacity.

Additionally, it was found that none of the reviewed previous literature on mortar joint detection techniques have been developed with the same level of masonry noise and distractions as seen on masonry arch bridges. Therefore, ways of detecting mortar joints in noisy masonry arch bridge images, where the mortar is often obscured by paint, dirt or vegetation have been explored.

5.1 Deterministic pattern recognition

A new method to increase the accuracy of mortar joint mapping by exploiting the regular arrangement of bricks in masonry walls has been proposed. In this way, false mortar joint detections can be discounted and missed ones added, increasing the resilience of mortar joint detection to image noise. The method employs a simple edge detection mask to identify potential mortar line locations, using this mask to determine the brick spacing

pattern, from which mortar line locations can be determined. This methodology has been developed and executed in the Matlab language and computing environment (MathWorks, 2018).

An overview of the developed method is outlined in Figure 5-1. The method consists of three stages. Firstly, a pre-processing stage is used to determine parameters of the brickwork imaged. Next, the bed joint (horizontal mortar line) locations in the image are determined. The image is then divided up into brick courses determined by the bed joints detected. Finally, for each brick course, head joint (vertical mortar line) locations are determined using a similar process as used for determining bed joint locations. The method developed assumes orthorectified images, i.e., that the images show masonry with no perspective distortion, as per the images developed in the dataset in Chapter 4. Figure 5-2 shows the main output stages involved in the process. These stages are not in process order, as bed and head joints are processed sequentially. These stages are detailed in the following sections.

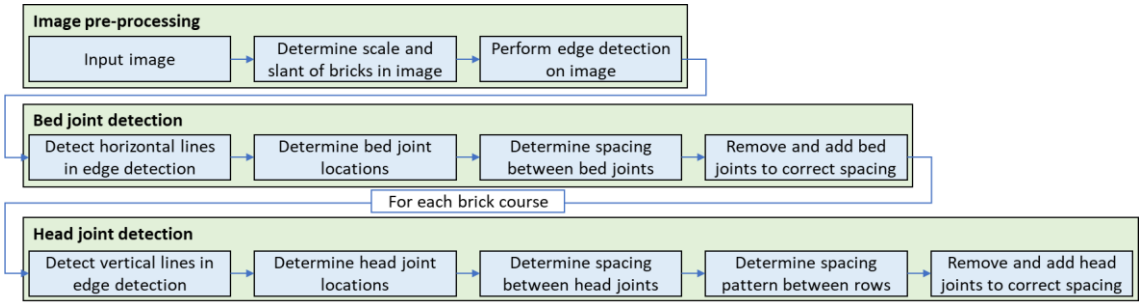


Figure 5-1: Workflow for mortar line detection methodology

Figure 5-2: Stages of mortar line detection: a) input image data, b) edge detection, c) horizontal and vertical straight-line detection, d) mortar line plotting, e) pattern detection and mortar line detection correction

5.1.1 Image pre-processing

Pre-processing involves determining the brick size relative to the image resolution and the orientation of the masonry in the image, so that the factors later used for feature detection can be adjusted to account for the image conditions. Although it is assumed that

the input image data will be derived from a textured 3D geometric model of a bridge, this is a necessary step as brick orientation can vary, for example on skew arches, and the bricks used might not be of a standard size. An example of the utility of such an approach is demonstrated in Figure 5-3, where the image has initially been rotated to simulate masonry at an angle. The masonry angle has been automatically detected, such that the mortar joint detection process can still work effectively.

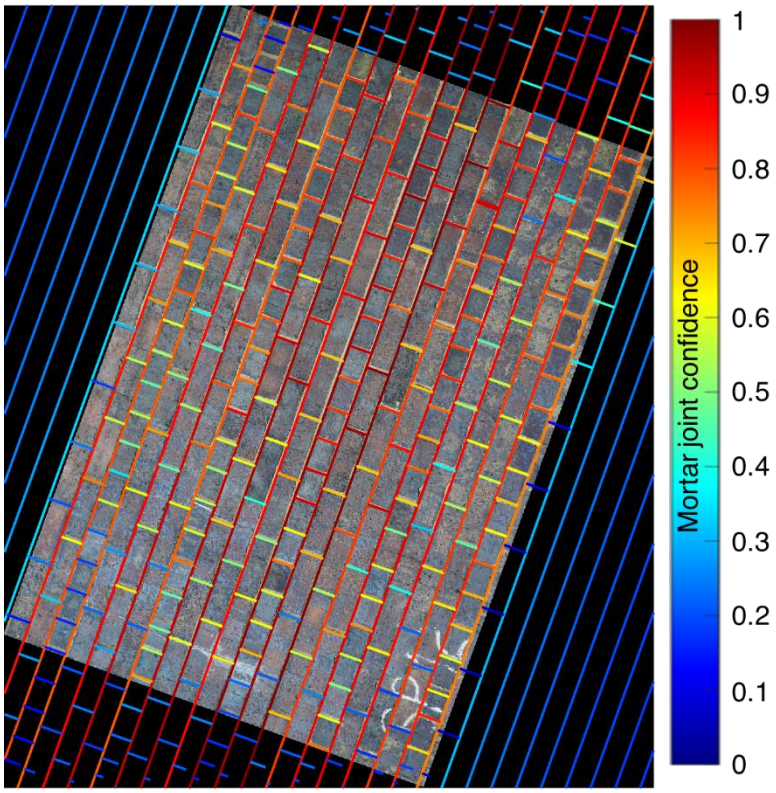


Figure 5-3: Mortar joint detection on a rotated input image

These size and orientation aspects are determined from the orientation and spacing of bed joints in a sample region of the image. Bed joints in the sample region are scanned for using the same methodology as is later used for bed joint detection across the whole image. For this, the masonry is assumed to have an initial arbitrary image scale which is iteratively updated. Properties of the detected bed joints in the sample window are measured to ensure that concurrent bed joints have been detected and not random noise. These properties are the number of straight lines detected in the image, and the number of orientations these are at, the number of clearly separated bed joints detected, and the consistency of the spacing between the detected bed joints. If these properties suggest the detection is poor, then the sample region size is increased and/or the arbitrarily assumed masonry image scale is changed. The sample region is then scanned again. Otherwise, the

detected bed joint spacing and the rotation of the bed joints relative to the horizontal are recorded and set as the properties of the masonry in the image.

5.1.2 Bed joint detection

Image intensities rather than colour thresholding were chosen as the feature for extracting mortar joints. This was chosen as in some masonries, particularly older masonry such as on arch bridges, there is no clear colour differentiation between mortar and bricks due to coverings of for example dirt or paint. Therefore, Sobel edge detection is performed to detect the changes in image intensity experienced at bed joints. In many masonry arch bridge images, bed joints are much more prominent in some regions than others due to noise on the bridge surface. Therefore, the image is divided into smaller regions, each only a few bricks in size, and edge detection is performed on each image region separately. This means that a tailored threshold for edge detection can be set locally for each image region, based on the detected intensity of the bed joints. Figure 5-2b shows the edge-detected output of a masonry image.

The Hough transform is used to detect regions of straight lines in the edge detection mask. This finds straight lines in an image using a voting procedure to select the most likely candidates. To account for the local variation in the prominence of bed joints, the Hough transform is performed separately on the same image regions used for the edge detection. The straight lines detected are then filtered so that only those forming a part of a bed joint are retained. This is done by removing all straight lines which are not orientated horizontally relative to the masonry angle determined pre-processing the image. These detected straight horizontal lines are shown in green in Figure 5-2c.

The detected horizontal straight lines are split into groups based on the bed joint they represent. This is done by grouping straight lines with similar vertical coordinates relative to the masonry angle. Grouping is achieved from the peaks in a kernel distribution fitted to the location and weighted by the length and confidence assigned to each of the straight lines, as illustrated in Figure 5-4. The bandwidth of the kernel distribution is set based on the scale of the brick size relative to the image determined in pre-processing.

The horizontal spacing between bed joints is assumed to be constant in masonry due to the constant height of brick courses. The spacing between each detected bed joint and its neighbour is calculated, and a kernel distribution approximation is used to group bed joints with similar spacing. This process is comparable to that in Figure 5-4, with the weighting for each spacing being assigned by the confidence in the positioning of the bed

joints. The bed joints spacing is assumed to be the peak in the kernel distribution corresponding to the minimum spacing, except when the prominence of this peak is not of sufficient magnitude. This is because if bed joints are missed, then there will also be peaks representing larger spacing, and if a bed joint is falsely identified it is likely to cause a low prominence peak representing a small spacing.

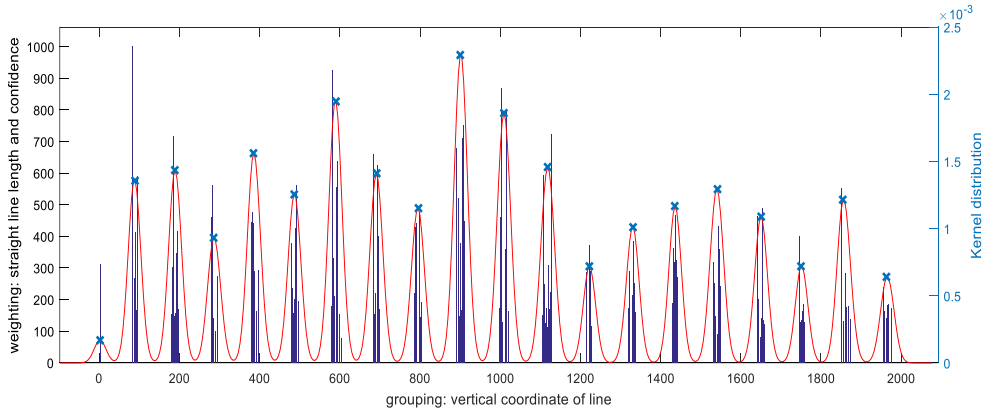


Figure 5-4: Fitting of kernel distribution to group straight lines into the bed joints they form

If the spacing between one bed joint and both its neighbours is not correct, i.e., not approximately a multiple of the determined bed joint spacing, then that bed joint has been incorrectly identified and is removed, otherwise the bed joint is correctly positioned. At the edge of the image where the bed joint only has one neighbour, if the spacing between it and its next correctly positioned neighbour is not correct, it is incorrectly positioned and removed. Extra bed joints are added and distributed where the spacing between bed joints is greater than the determined bed joint spacing.

Bed joints are then plotted by fitting a linear polynomial equation to the length-weighted positions of straight lines in each group. As bed joints run parallel to one another, the gradient of each bed joint is then weighted by those surrounding it using an exponential decay process. The exponential decay factor is assigned based on the confidence in the positioning of the bed joint, such that the gradient of a bed joint with low confidence will be based heavily on those surrounding it and less so on its own initially calculated gradient. Conversely for a bed joint with high confidence, the value of its initially calculated gradient will carry a large weighting in determining its final gradient. The plotted bed joints detected are shown in green in Figure 5-2d.

5.1.3 Head joint detection

The image is divided into brick courses using the determined placement of the bed joints, and head joints are detected in each brick course separately. The process for head joint detection for each individual row is then comparable to that for bed joint detection. Straight lines detected by the Hough transform are filtered to only retain lines which are oriented perpendicular to the bed joints surrounding the brick course being processed. These vertical straight lines are shown in blue in in Figure 5-2c. The detected lines are grouped by approximating a kernel distribution to their location comparable to the process in Figure 5-4 for bed joints, though this time based on the horizontal coordinate relative to the masonry angle.

Deviating from how the bed joint orientations were determined, head joints are assigned a gradient perpendicular to the surrounding bed joints. To determine the centroid of the head joint, attention is focused on the spacing of the grouped straight lines. If the horizontal coordinates within a group appear to fall into two subgroups, then this suggests that straight lines have been detected on both sides of the head joint, i.e., lines have been detected on the interface between brick and mortar on both the left-hand side and the right-hand side of the mortar. The two subgroups are detected by approximating a kernel distribution to the horizontal coordinates of the straight lines within each head joint group again, this time with a much smaller bandwidth. Where there are two peaks in the kernel distribution, lines have been detected on both sides of the head joint, and these head joints are classified as doubly defined (see Figure 5-5). In this case, the head joint centroid is the mean of the two peaks and its thickness can be determined from the distance between the peaks. If a single peak is detected in the kernel distribution, lines have only been detected on one side and the joint is classified as singly defined (see Figure 5-5). In this case, the head joint centroid is the mean of the centroid coordinates of the straight lines defining the head joint. If there are more than two peaks in the kernel distribution, then it is assumed that there is noise in the image and the head joint is deleted.

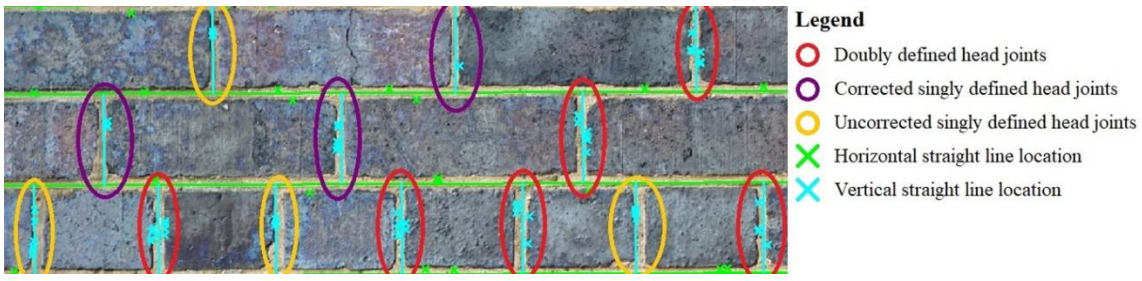


Figure 5-5: Correction of singly defined head joints

The spacing between head joints is identified for each brick course individually in the same way as for bed joints. This spacing is used to identify whether each course is a header course (small face of the brick visible so a small spacing between head joints) or a stretcher course (large face of the brick visible so a large spacing between head joints). For this, courses are grouped based on their type, by approximating a kernel distribution to the head joint spacing for the different courses, weighted by their spacing confidence. By marking the points where the courses change type, the number of consecutive occurrences of each course type is recorded for all incidences across the image. For each type of course (header or stretcher), the most common sequential number of occurrences of the course type weighted by the confidence in the spacing of the individual courses in that incidence is recorded. These numbers of occurrences are used to build up the predicted bonding pattern between the different brick courses. If the measured spacing between head joints of a brick course with low confidence disagrees with the predicted bonding pattern, its spacing is changed to suit the predicted pattern.

Where head joints are singly defined, their location is corrected so that they represent the middle of the joint, not one edge. This is done by shifting the horizontal coordinate of the head joint by half the average mortar thickness calculated from the doubly defined head joints. The direction to shift the head joint is determined by the space to the next doubly defined head joint in each direction. This is decided by whether the space is greater than or less than a whole number of multiples of the head joints spacing value. If the doubly defined head joints to the left and to the right of the singly defined head joint both agree with the direction it should be moved in, then it is moved in that direction. Otherwise, it is left in place. This process is illustrated in Figure 5-5. Next, all the head joints are added and removed based on the spacing between them, in the same way as for bed joints. Position corrected head joints are shown in red in Figure 5-2e.

5.1.4 Detection confidence

As part of the methodology, a measure of confidence for each plotted mortar joint is calculated from the confidence of each step required for determining its location. As discussed above, these confidences are used in determining whether each mortar joint has been plotted correctly.

The different regions in the image are given a confidence weighting based on the number of different orientations of straight lines that were detected by the Hough transform. Where more orientations are detected, this suggests that not all the straight lines represent

mortar lines and confidence is reduced. The straight lines detected are assigned a confidence based on the confidence of the region of the image where they are located.

The confidence of mortar lines is based on the confidence, the number, and the length of the straight lines they are plotted from. For bed joints, the confidence is also based on the spread of straight lines across the length of the mortar line as a greater spread means that the extrapolation of the line is less. For head joints, the confidence is also based on the confidence of the two bed joints surrounding the brick course, as misplacement of these would result in a much less robust detection of head joints for that brick course. Additionally, doubly defined head joints are marked with greater confidence than singly defined, as there is more certainty over their positioning. Where a mortar line is added based on spacing, its confidence is based on the surrounding mortar lines, but diminished as the evidence in the image suggesting its position has not been detected. The confidence of the spacing between mortar lines is defined by the confidence of the mortar lines.

5.1.5 Testing the methodology

Features of the methodology have been evaluated on the stitched image of a complete pier of Hertford Viaduct discussed in Chapter 4.4 and depicted in Figure 4-3. The masonry in this pier was 39 stretcher bricks in width and 84 brick courses in height. It is relatively clean, though shows some regions of vegetation occlusion as well as low contrast caused by staining of the brick work. The accuracy of the methodology was evaluated against a manually created mask depicting the locations of all the mortar lines, the creation of which is described in Chapter 4.6.

Firstly, the effect of the resolution of the input image on the detection accuracy was determined. This was done by taking the test image and reducing its resolution. The methodology was tested on image resolutions of between 0.15 and 3.02 pixels per mm of masonry surface. The resulting accuracy is shown in Figure 5-6a. This demonstrates that the resolution of the input image has negligible effect on the accuracy of mortar line detection as the pre-processing stage detects the scale of the masonry relative to the image and parameters are adjusted to suit. Where the image resolution gets below about 0.5 pixels per mm, the accuracy of detection diminishes. At this resolution, a 10mm mortar line would only be represented by 5 pixels making it much harder to distinguish. Prior to the detection accuracy diminishing, the average head joint position error is 1.25mm on the masonry surface. This is the average distance between where head joints have been detected and their actual location depicted on the manually annotated image mask.

The effect of the number of bricks per image on detection accuracy was also evaluated. This is thought to be significant as the methodology works by detecting the mortar line patterns, so having more bricks per image would give more opportunity to accurately determine the pattern. For this, the panorama image was used at a resolution of 1.5 pixels per mm and segmented into smaller images. The aspect ratio and masonry resolution of the segmented images were kept constant. The resulting accuracy is shown in Figure 5-6 in b and c. As discussed in Chapter 4.4 and shown in Figure 4-4, the stitched panorama was found to have a slight inconsistency in one part where the stitched images did not quite align. This resulted in a slight nonlinearity of the imaged bed joints. Additional tests were conducted avoiding this region of the image and are shown in red.

Figure 5-6b shows that as the number of stretcher bricks per image gets smaller, then the accuracy of bed joint detection improves. This improvement suggests that the error in bed joint detection is a result of assigning a straight line to mortar joints which are very long, and not necessarily perfectly straight across the whole pier length. As the number of stretcher bricks per images reduces further still, there is greater variation in the accuracy of detection. This suggests that the detection becomes more susceptible to noise. The optimal width of masonry imaged was about 15 stretcher bricks wide. Figure 5-6c shows that above about 17 stretcher bricks in width, there is insignificant effect of the width of masonry imaged on the accuracy of head joint detection. Below this however, the process was, in a few cases, susceptible to noise.

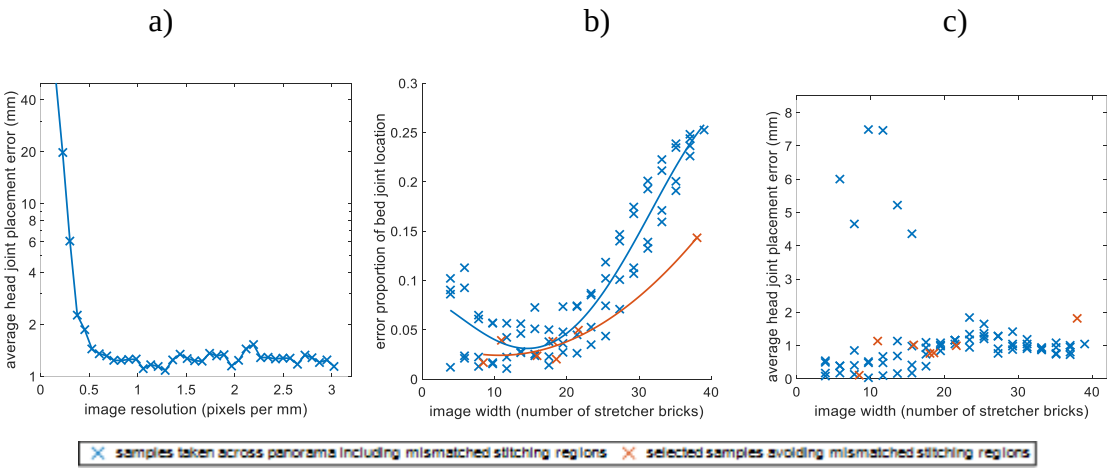


Figure 5-6: Mortar line accuracy: a) head joint placement varying image resolution, b) bed joint placement varying image width, c) head joint placement varying image width

The methodology has also been tested on the two other stitched viaduct pier images described in Chapter 4.4. Small portions of all three viaducts are shown in Figure 5-7. These images show significant variation in the image conditions. Figure 5-7a shows a case where pattern recognition is used to successfully identify all mortar joint locations despite occlusion by vegetation. Figure 5-7b demonstrates a difficult detection condition where the masonry is painted. There is therefore no colour distinction between the bricks and the mortar, meaning that the edge detection is utilising only textural information. Examining the edge detection output (Figure 5-7b, middle row), there are many places where it is not clear where the mortar lines should be positioned. However, the final joint detection is successful. This demonstrates the advantage of detecting the masonry pattern to enhance detection, while putting limits on extrapolating the pattern given it can change across larger structures.

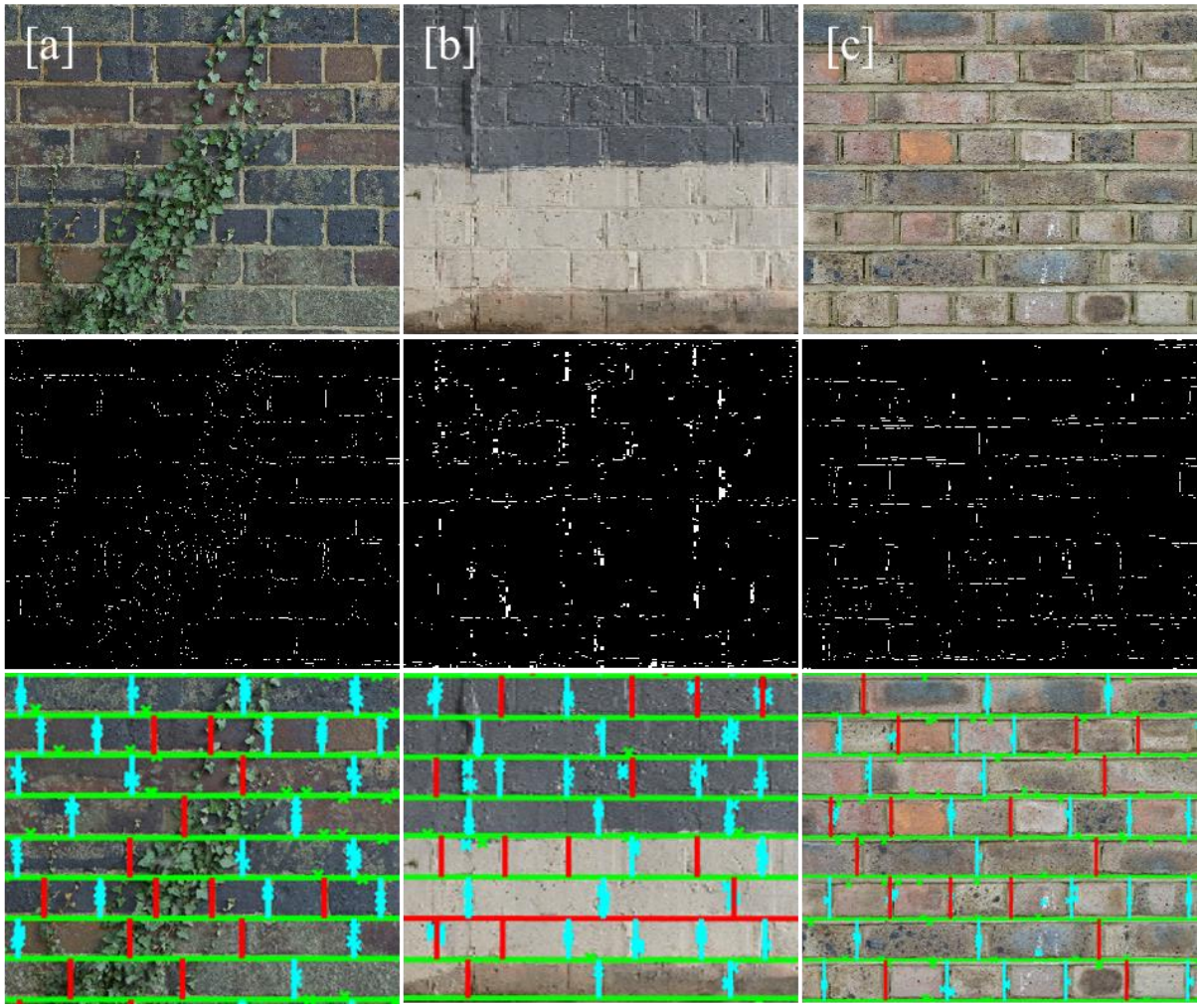


Figure 5-7: Closeup images from testing on different bridges showing input image, edge detection and output, for: a) Hertford Viaduct, b) Peterborough Viaduct, c) Chelmsford Viaduct

The precision and recall of the methodology have been assessed using these images of different viaduct piers. The masonry imaged in each case was 19 stretcher bricks in width and 17 brick courses in height. Precision, defined in Equation (5-1), measures the proportion of the points detected that really are mortar lines, and recall, defined in Equation (5-2), measures the proportion of mortar lines that have been identified.

$$Precision = \frac{number\ of\ true\ positives}{number\ of\ true\ positives\ +\ number\ of\ false\ positives} \quad (5\text{-}1)$$

$$Recall = \frac{number\ of\ true\ positives}{number\ of\ true\ positives\ +\ number\ of\ false\ negatives} \quad (5\text{-}2)$$

For each image case the performance of the output was assessed for: edge detection, plotting of mortar lines without pattern detection and correction, and plotting of mortar lines with pattern detection and correction. These outputs are illustrated in Figure 5-2 in b, d, and e, respectively. The results in Table 5-1 show that when compared to edge detection alone, using the proposed methodology reduces the subjectivity of mortar joint detection. Where image conditions are respectable, as in Hertford viaduct, edge detection can quite accurately detect the location of mortar joints. However, where image conditions deteriorate, as in Peterborough viaduct, the edge detection output is not very accurate. By plotting mortar lines between the detected points and detecting their pattern, the detection accuracy is greatly improved. In all cases the proposed methodology improved the mortar joint detection performance of the edge detection used as its input.

Table 5-1: Precision and recall of the methodology tested on different viaduct images

Viaduct	Edge detection		Plotting mortar lines without pattern detection and correction		Plotting mortar lines with pattern detection and correction	
	Precision	Recall	Precision	Recall	Precision	Recall
Hertford	95.10%	80.43%	97.47%	95.93%	96.81%	96.36%
Peterborough	68.53%	42.65%	79.74%	62.61%	87.17%	86.23%
Chelmsford	79.88%	68.11%	98.89%	83.61%	95.01%	92.92%

As discussed in Chapter 2.4 in the literature review, Riveiro et al. (2016) have crafted a methodology for detecting joints in masonry based on a similar principle of using edge detection for initially detecting bed joints, followed by head joints in each brick course individually. However, this methodology was designed for detecting dry joints rather than mortar joints, which are likely to show a larger surface discontinuity. Additionally, this methodology was proposed for identifying individual bricks to suggest markers for a watershed segmentation operation rather than to detect the actual joints. It is therefore much less developed, particularly lacking any pattern detection features.

The developed methodology is a hardcoded methodology, and as a result has manually set factors and thresholds to tune the methodology to detect and interpret features in masonry images for determining mortar joint positions. In Figure 5-7 and Table 5-1, the methodology has been tested on different bridges depicting a range of masonry conditions, suggesting to a degree that the methodology is capable of generalising. To improve the generalisability of the methodology across multiple different masonry conditions, these factors and thresholds have been tuned on a series of 12 masonry images, with a performance objective of maximising the F_1 score of mortar joint detection. The F_1 score is the harmonic mean of the precision and recall, defined in Equation (5-3). Therefore, the objective is to maximise both precision and recall.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{5-3}$$

The 12 masonry images have been selected from the dataset of annotated perspective corrected images of viaduct data described in Chapter 4.6, chosen for depicting a wide range of masonry conditions. 35 tuneable factors from the methodology have been identified, and these factors have been individually tested on each of the 12 masonry images, to establish its effect on the performance objective. As these factors have been tested individually, this is not a complete optimisation of the process, as the interdependence of the individual factors has not been examined. This is because from a research perspective, the effect of each factor on the performance has a greater interest than a complete optimisation.

The output of varying these 35 factors individually on the 12 test images has been presented in Appendix 1. From these figures, it is apparent that some of the factors have an obvious trend across the mortar joint detection performance of all the test images, resulting in an optimum value. Secondly, some factors have insignificant effect on the mortar joint detection performance in the test images, showing constant scores across the

trialled factor values. Thirdly, some factors result in a different optimum for different test images, making setting an optimum problematic. An example of this third case is shown in Figure 5-8 which shows the performance output for one of the factors studied in Appendix 1. These un-tuneable factors limit the generalisability of such a hardcoded methodology, showing how tuning a factor to improve the performance on one image, could result in a decrease in performance where the masonry presents slightly differently. Machine learning, and more so deep learning approaches improve generalisability through having the potential for incorporating many more factors for determination, as well as optimising on many more image samples. Such approaches will now be explored for detecting mortar joints in arch bridge masonry.

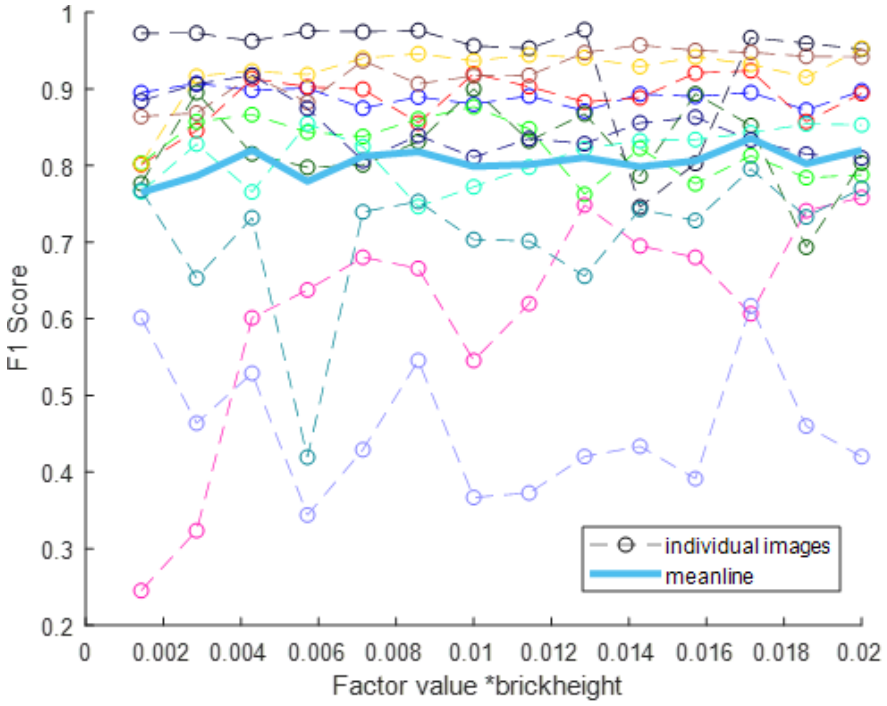


Figure 5-8: Effect of the standard deviation of the Gaussian filter used to smooth the image prior to edge detection on the F1 score of mortar joint detection in different test images

5.2 Deep learning

A Deep learning rather than a machine learning classifier has been used, as studies discussed in the literature review in Chapter 2.3 have shown them to be much more robust to image noise, which is prevalent in images of masonry arch bridges. Specifically, Convolutional Neural Network (CNN) based approaches have been applied, as these are best suited to image-based data. CNNs work by layering filters which are convolved across the image. These filters are each optimised to pick up different features in the

image. In the deterministic mortar joint detection method described in Chapter 5.1, just an edge detection filter was used meaning edge features were the basis of the methodology. However, with deep learning many different features can be used simultaneously as a basis for classification. In a CNN, the higher layers of the network extract more involved or more specific features from combinations of these lower layer features.

5.2.1 Sliding window methodology

Initially a sliding image window based deep learning mortar joint classification method has been trialled. For this, the masonry image is divided into many smaller samples, based on the sliding window principle. An overlap between adjacent window samples has been set as half the image window size, such that image windows with a size of 30 pixels will have an overlap of 15 pixels. This means that each part of the masonry image will appear in 4 separate image windows. An image window is defined as depicting a mortar region if at least 1/3 of the pixels in the central region of the image window are mortar pixels. The image window central region is defined such that all the pixels in the image will be in the central region of exactly one image window.

Classification is performed using the GoogleNet Inception v3 architecture (Szegedy et al., 2016) as it is one of the best performing models against the ImageNet classification benchmark. Only much more computationally expensive models have achieved slightly better performance (Canziani et al., 2017). This publicly available model has been pre-trained using the 1000 classes and 1.4 million images of the ImageNet dataset (Deng et al., 2009). In this case, the network has been used as a feature extractor, such that the lower layers which have been pretrained are retained, and only the top layer, the Softmax classifier has been randomly initialised and retrained on the mortar joint data, with the weightings of other layers fixed. Therefore, the features that the convolutional layers of the network have learned from pre-training on the ImageNet data have been transferred to this new domain. Training has been performed with a learning rate of 0.01 across 4000 image window samples using the Tensorflow machine learning platform (Abadi et al., 2016).

This classification has been trained and tested on stitched images of the piers of Peterborough, Hertford and Chelmsford viaduct as described in Chapter 4.4, though the image data has been scaled such that it has a resolution of approximately 65 pixels per brick course. As the same dataset has been used for both training and testing there is no

test of generalisation of the classification, though the limited training samples used in comparison to the training dataset size means that the classifier will not have seen most of the data and is very unlikely to have overfitted to the data it was trained on. The method presented here is therefore more to test the concept rather than develop a finalised classifier.

Different image window sizes of between 10 and 35 pixels have been used, to determine the effect of the amount of masonry imaged in each window on the ability to classify mortar regions. As the image resolution is approximately 65 pixels per brick course, each image window therefore depicts less than a whole brick of masonry for all the window sizes tested. The performance of the classifier for these different image window sizes and on the different viaduct piers tested are shown in Figure 5-9. The performance of classification improves as the image window size gets larger.

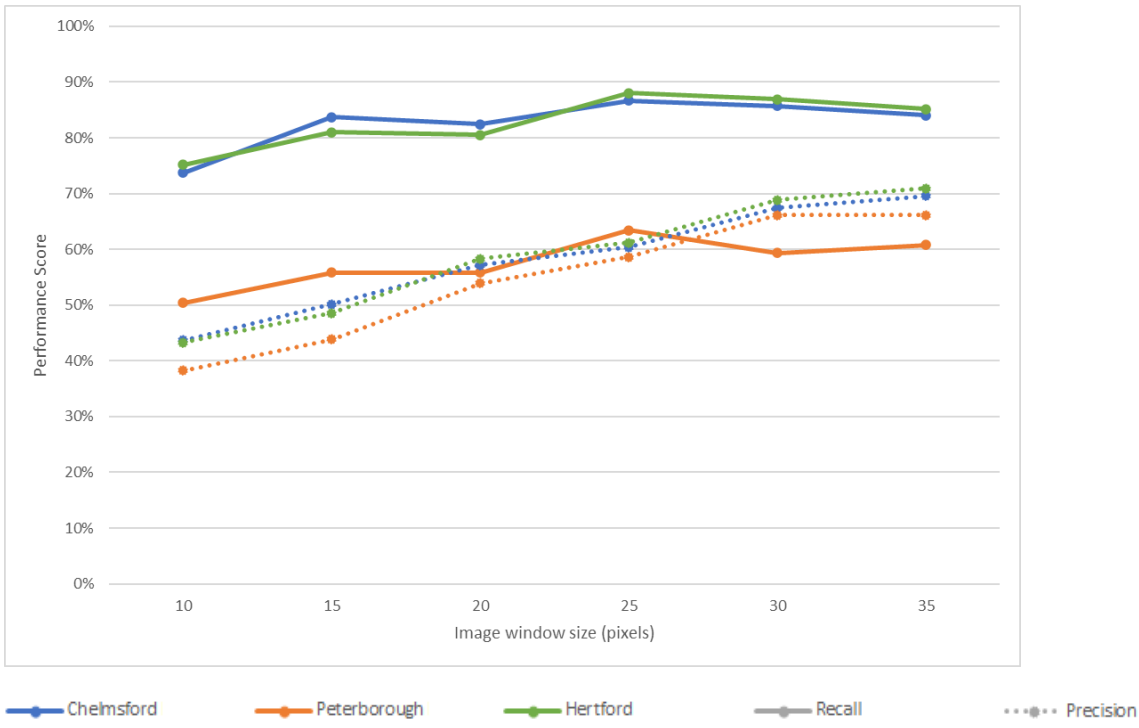
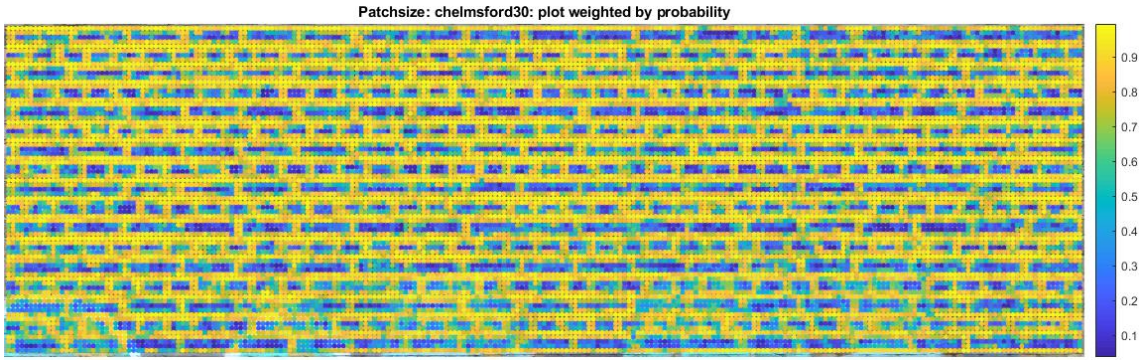


Figure 5-9: Image window mortar joint classification performance for different image window sizes and viaduct piers

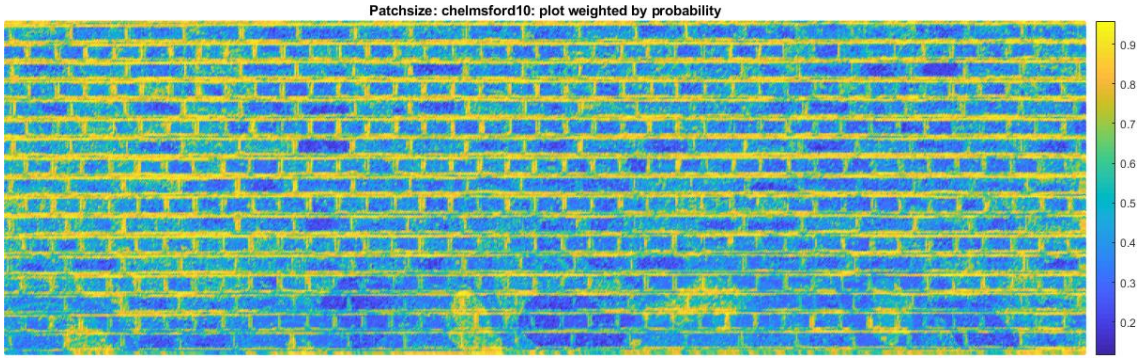
However, performance is no better than the edge detection performance shown in Table 5-1, though if anything it is slightly more consistent between the different bridge surfaces. Image window wise segmentation also means there is a lower granularity of classification when compared to edge detection which is pixel wise. This is especially prominent for the larger image window sizes as shown in Figure 5-10, so although they offer greater classification performance, their classification is less useful. The better classification

performance on the larger image window sizes is most likely due to there being more information in each image window to base a classification on. Even with the larger image window sizes, given these are still smaller than a single brick, classification is still based solely on the mortar joint texture and interface with no consideration of the mortar pattern. This is apparent in Figure 5-10 where some of the brick sections are coloured much lighter than others, suggesting confusion with mortar. This is especially prevalent in areas where there is yellow graffiti suggesting classification is to an extent being based on pixel colouration.

a)



b)



c)

Figure 5-10: Image window mortar joint classification for image from Chelmsford viaduct pier with image window size of: a) 30 pixels and b) 10 pixels. c) depicts the tested image

5.2.2 Semantic methodology

The disadvantages with an image window classification discussed above suggest a semantic, pixel-wise classification is more appropriate. This enables a higher granularity of mortar joint classification location as well as the ability to train and test on larger portions of images such that a classification can be based on more than just localised surface texture – for example, mortar pattern information can be considered. In a non-semantic, image-window based CNN classifier, location information within the image is preserved throughout the convolution layers, and only lost at the fully connected layers at the end where the class classification is conducted. Semantic classifiers therefore have similar architectures to whole image CNN classifiers, both being based on convolution layers due to their suitability for image data, though with semantic classifiers the final fully connected layers are replaced, creating a fully convolutional structure. Typically, the first part of the network is the same as in a CNN for whole image classification. This uses convolution layers coupled with down-sampling layers to encode the image into a low-resolution feature map containing high-level data. With semantic classifiers, further convolution layers coupled with up-sampling layers are added, to then decode a high-resolution semantic output.

There are several different semantic segmentation algorithms employing different structures with similar principles. In this work, a Pyramid Scene Parsing Network (PSPNet) has been used. This was chosen as it is one of the most well recognised and highly cited image segmentation algorithms having won the ImageNet Scene Parsing Challenge 2016 (Zhao et al., 2017). The PSPNet is built around the Pyramid Pooling Module, outlined in Figure 5-11. This takes the feature map from the last convolutional layer of a CNN encoder and pools it at different sizes, each then being passed through a convolutional layer and up-sampled back to its original size. These feature maps are then concatenated with the original feature map which then is passed on to further convolutional layers for decoding into a semantic output. This means that information from the feature map is captured at various levels of coarseness, increasing the size of the receptive field of the output. This structure therefore makes a PSPNet ideal for learning the global information from images to aid in local level predictions as well as being able to capture objects of varied sizes in images. This means it should be able to learn the global mortar pattern as well as the local mortar textures.

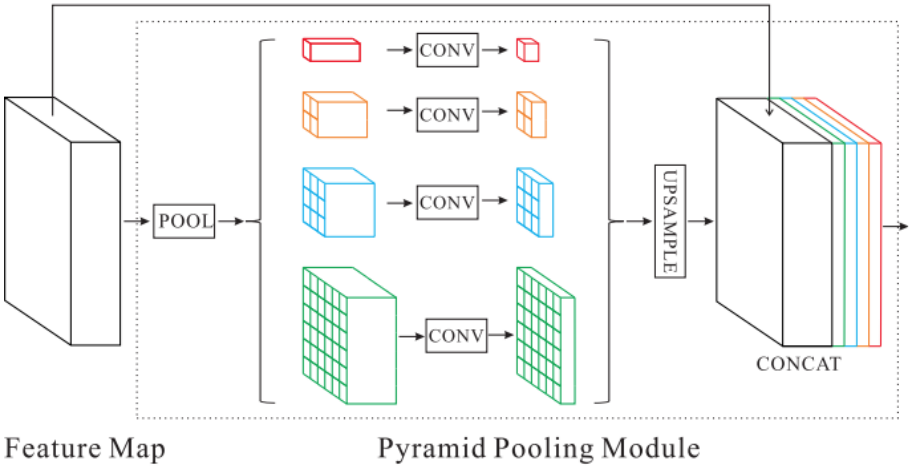


Figure 5-11: Pyramid Pooling Module as used in the PSPNet architecture (Zhao et al., 2017)

The PSPNet used has been implemented with a 50 layer deep ResNet (He et al., 2016) CNN as its encoder, and has been pre-trained on the ADE20K dataset (Zhou et al., 2017). This is a dataset of 20,000 scene centric images annotated at a pixel level with 150 categories. These categories include surfaces, such as roads and grass as well as objects such as buildings and people. The model has then been fine-tuned on brick and mortar data. Unlike with a classification-based CNN, to fine-tune a semantic, fully convolutional CNN only one training stage is needed. This is because the fully connected layers that are replaced when fine-tuning a classification-based CNN which necessitate the additional tuning stage are removed from a semantic CNN. The fine-tuning process used is described as follows.

The data used for training and testing are the first and second datasets, respectively, described in Chapter 4.6. In the bridge location map in Figure 4-2, this is data from those bridges with blue markers (predominantly viaducts) for training images, and data from those with red markers for testing. Therefore, data for training and testing has been acquired from separate bridges, to evaluate the generalisability of the method. Just the pixel-wise mortar joint location mask and the base image data have been used, with the defect location masks discarded. The images in both datasets have been segmented into square image patches with a size of 500 pixels. As the images in the datasets have a resolution of 155 pixels per brick course, this means that several brick courses are included in one patch as shown in Figure 5-12.

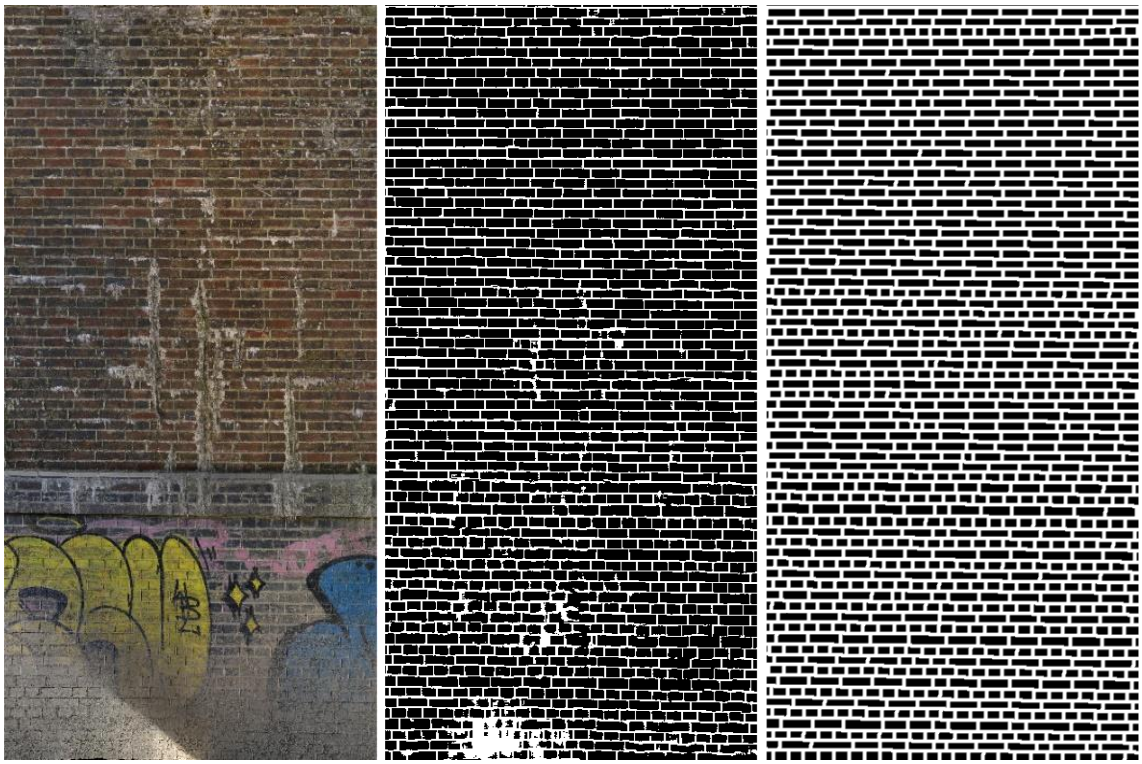
Figure 5-12: Example of image patch size used for training and testing

The image patches have been extracted from the dataset images such that there is a 50% overlap between each image patch. This means that each pixel in a dataset image will appear in 4 separate image patches. This is done because it is noted by Yang et al. (2018) that with a semantic segmentation approach, the accuracy of crack segmentation closer to the image borders declines, so by overlapping patches it means that no pixel is only tested at the edge of an image patch. This has resulted in a training set of 18,480 images, of which 1,614 have been used for validation, and a testing set of 27,123 images. Fine-tuning training has been performed using the Keras machine learning framework (Chollet and Others, 2015). A Root Mean Squared Propagation (RMSProp) optimizer (Hinton, 2012) with a learning rate of 0.0001 across 20 epochs of the training data has been used, with all weights across the network made trainable.

5.2.2.1 Results

As discussed, the developed semantic classifier has been tested on the second dataset described in Chapter 4.6, i.e., those bridges with red markers in Figure 4-2. This dataset contains complete 2D textures for all the surfaces of 3 different underline bridges, developed through photogrammetry and UV mapping. A section of one of these bridges is shown in Figure 5-13 where the output of the semantic segmentation is shown in b. It is apparent that the semantic segmentation has effectively ignored image noise in the form of, shadowing, light patches, block fill painting, thin line painting, steps in the brick courses, efflorescence and staining, gouges, and thick dirt deposits. In almost all locations, mortar joint locations have been accurately classified except in a patch towards the bottom of the image where the surface texture is covered in thick dirt deposits, which has resulted in confusion. However, the mortar joints are almost impossible to manually discern from the image in this area. This therefore shows the potential benefit in applying a pattern recognition procedure to a semantic output. This has been done in Figure

5-13(c), where the erroneous mortar joint region from the semantic output has been corrected. For this, the same methodology as used before in Chapter 5.1 is applied. Here however, the semantic methodology segments the whole width of the mortar joints rather than both edges as before with edge detection. Therefore, the semantic output has been used directly to detect the position of mortar joints using a kernel distribution as in Figure 5-4.

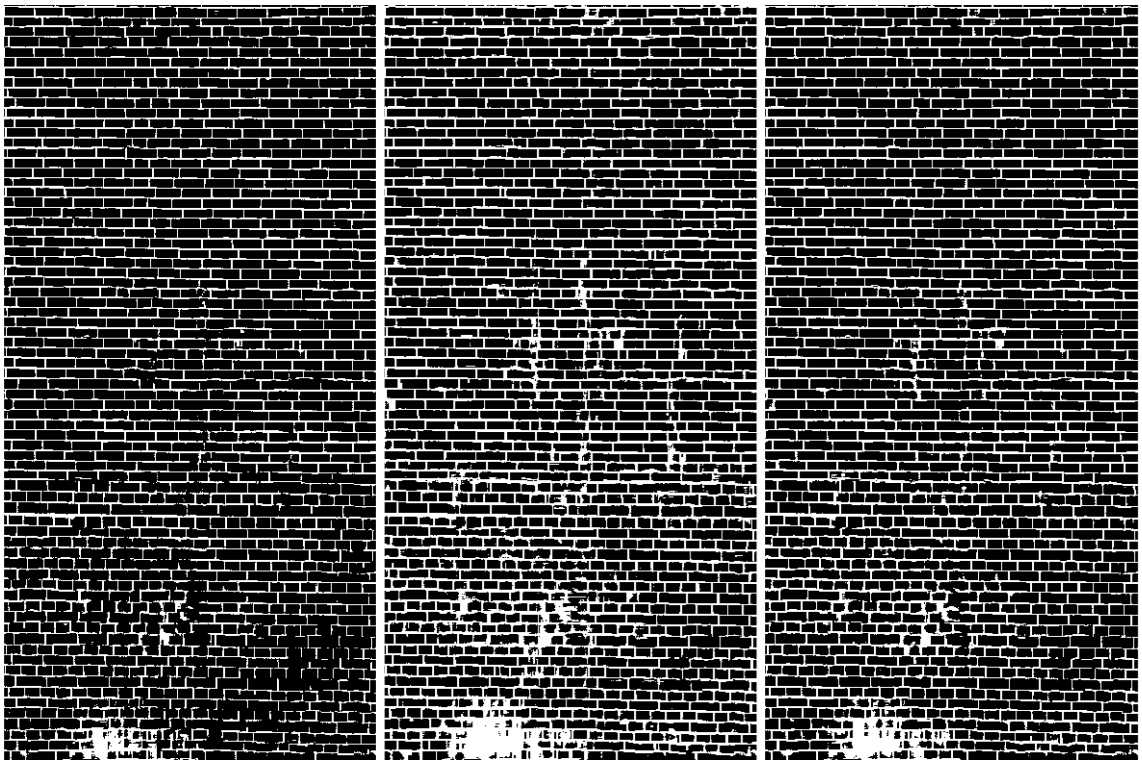


a) b) c)

Figure 5-13: Semantic segmentation output for one section of Smythes Farm Bridge (SBR 12) depicting part of the arch barrel and pier showing: a) input image, b) semantic segmentation output, c) deterministic mortar joint detection applied to semantic segmentation output

Above it has been discussed that the image patches into which the surface has been divided for testing overlap with each other by half their width, such that each pixel of the image surface is depicted in four different image patches. As the semantic classification of a pixel is slightly different based on where that pixel is in the tested image, this results in different potential outputs when stitching the tested patches back into their original texture. These different outputs are depicted in Figure 5-14. Firstly, Figure 5-14a depicts the minimum mortar scenario. This is where a pixel is only marked as a mortar pixel if all the overlapping patches agree that it should be a mortar pixel. Inversely, Figure 5-14b depicts the maximum mortar scenario, where a pixel is marked as a mortar pixel if any of

the pixels in the overlapping patches suggest it should be. Comparing these two images, the patch of incorrectly identified mortar is much smaller in the minimum mortar scenario in Figure 5-14a, though it is still present. Conversely, there are a few mortar joints that are missing in the minimum image Figure 5-14a, which are present in the maximum image Figure 5-14b. This suggests that although the minimum mortar strategy is better at avoiding false positive results, it also results in more false negative results. As discussed above, it has been suggested that the accuracy of a semantic classifier performs worse towards its edges. This is logical as the feature map is down-sampled and then up-sampled from the centre during classification. Therefore, a third patch stitching method is to discard the data towards the edges of each image patch, such that there is no overlap between the retained data in overlapping image patches. Therefore, the whole surface is represented through just the central regions of the individual image patches. This patch stitching method is depicted in Figure 5-14c. This results in a smaller false positive mortar detection patch than that for the maximum strategy in Figure 5-14b, yet it still correctly identifies the few missing mortar joints from the minimum strategy in Figure 5-14a.



a) b) c)

Figure 5-14: Semantic segmentation output method comparison for one section of Smythes Farm Bridge (SBR 12) depicting part of the arch barrel and pier showing: a) minimum mortar, b) maximum mortar, c) centre mortar

The performance of the different patch stitching methods has been compared in Figure 5-15. This is for the performance across all the surfaces on the three bridges in the testing dataset. As expected, from the previous qualitative analysis, the precision value is highest for the minimum mortar stitching method, where mortar regions are only marked if all overlapping patches determine a region to be mortar. Conversely, recall values are highest in the maximum mortar stitching method. The centre region mortar stitching method has the best balance between recall and precision scores, and hence the best F1 score.

Additionally, for the same set of images, the performance of the raw output of simple edge detection based on its ability to detect mortar joints has been calculated. This is the result of the edge mask only - before the deterministic method is performed. As edge detection only highlights the edges, i.e., the transitions between brick and mortar, these performance scores have been calculated through assuming 2 edge pixels should be present per mortar joint (both edges), whereas the semantic segmentation performance calculation assumed mortar pixels across the whole mortar thickness. As can be seen in Figure 5-15, the performance of edge detection was significantly worse than that of the deep learning semantic segmentation.

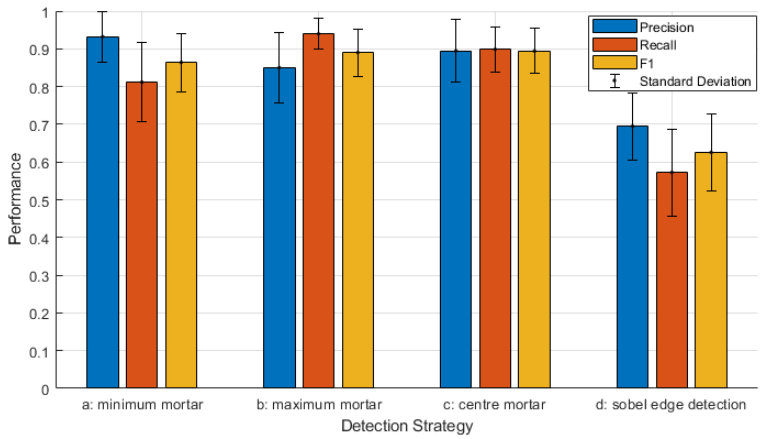


Figure 5-15: Performance metrics for different semantic prediction strategies

This worse performance suggests that the performance of the output of the deterministic methodology applied to the deep learning semantic segmentation should be better than when applied to an edge detection mask. This performance improvement is outlined in Figure 5-16. The performance of the deterministic mortar joint detection methodology using the edge detection mask input (i.e., Figure 5-16 d & h), is considerably worse than with all the other inputs in Figure 5-16, which are all deep learning semantic segmentation. This demonstrates that although the deterministic methodology can

improve the performance of mortar joint detection from its edge detection input, it benefits from having a more accurate input.

Also apparent in Figure 5-16, comparing e, f and g to a, b and c, there is a decrease in performance when the pattern correction methodology is applied as a part of the deterministic methodology compared to when the deterministic methodology is applied without correcting the pattern. The pattern correction methodology measures the spacing between detected mortar joints and adjusts the spacing of mortar joints which do not follow the expected pattern. However, as shown in Figure 5-16 i, j and k, the decrease in performance associated with applying the pattern correction methodology is reversed when only applying the methodology to vertical mortar joints.

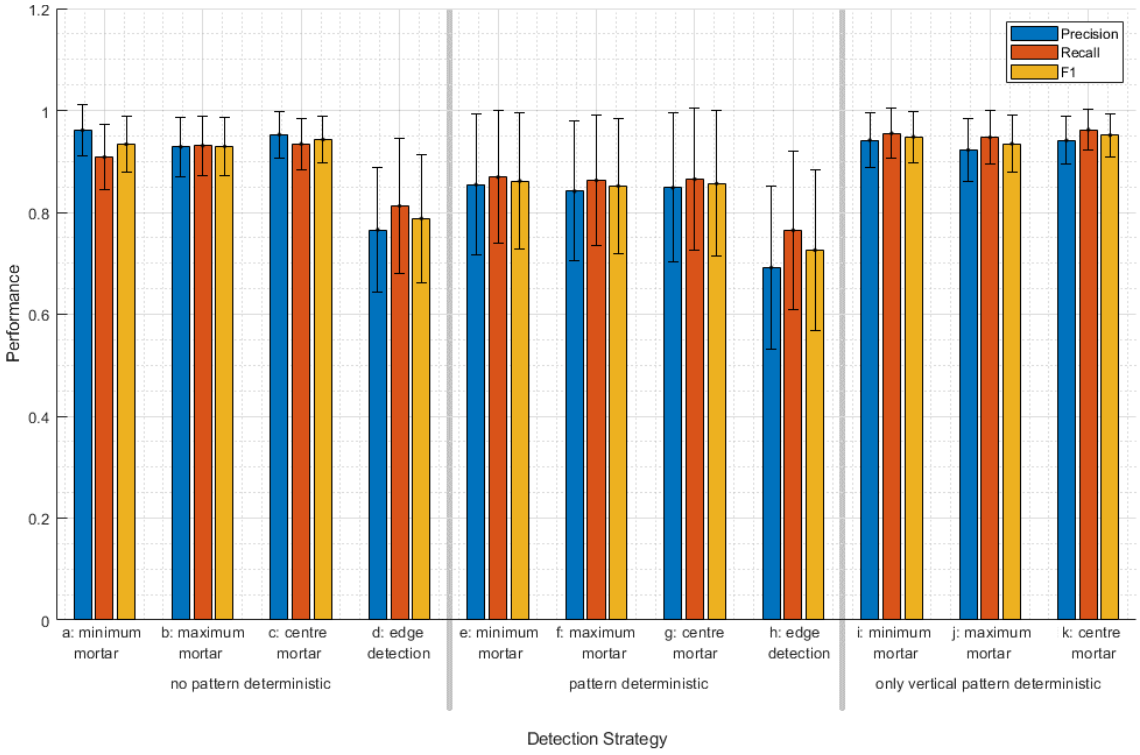


Figure 5-16: Performance metrics for different deterministic detection strategies

The probable cause of this is therefore irregular bedding planes in the images. For applying the deterministic methodology, the generated surface textures have been divided into images approximately 15 stretcher bricks wide. This was the ideal input image width determined in Chapter 5.1.5, giving the best balance between forcing a linear bedding plane mortar joint onto a potentially nonlinear alignment and ensuring there is enough data in the image to accurately position the mortar joint. However, this was determined from an image of a viaduct pier where the masonry was typically regular. In this case, all surfaces of the bridges have been tested, and therefore there are areas with non-regular

masonry, such as on the spandrel walls where a course of bricks generally runs along the circumference of the arch barrel. Additionally, there are areas where there is some distortion in the images from the UV unwrapping process due to an error in the calculated geometry.

Figure 5-17 breaks down the performance metrics into the performance of individual images. Here, each individual image is denoted by a grey line, and the mean is denoted by a black line. It can be seen, that in general, the performance improves with applying the deterministic methodology to the semantic segmentation output, and further improves with the vertical pattern recognition. One of the images following this trend is highlighted by an orange line, and this image can be seen in Figure 5-13. There are also a few images which do not follow such a trend, and two of these are highlighted by green and purple lines. These images can be seen in Figure 5-18 and Figure 5-19, respectively. Both images are from the wing-walls of Norton Road Bridge (SBR8) where the geometry of the structure has two dimensions of curvature, curving from being parallel to the spandrel wall to being parallel to the road, and curving from being sloped towards being vertical. Therefore, it is mathematically impossible to map the texture onto a 2D plane without either stretching or splitting. As curvature in the vertical sense is only a small amount, the texture has been fitted onto a plane though that has resulted in slight stretching and therefore nonlinear bedding joints, as is apparent from the images in Figure 5-18 and Figure 5-19. This has resulted in a misfitting of mortar joints using the deterministic method.

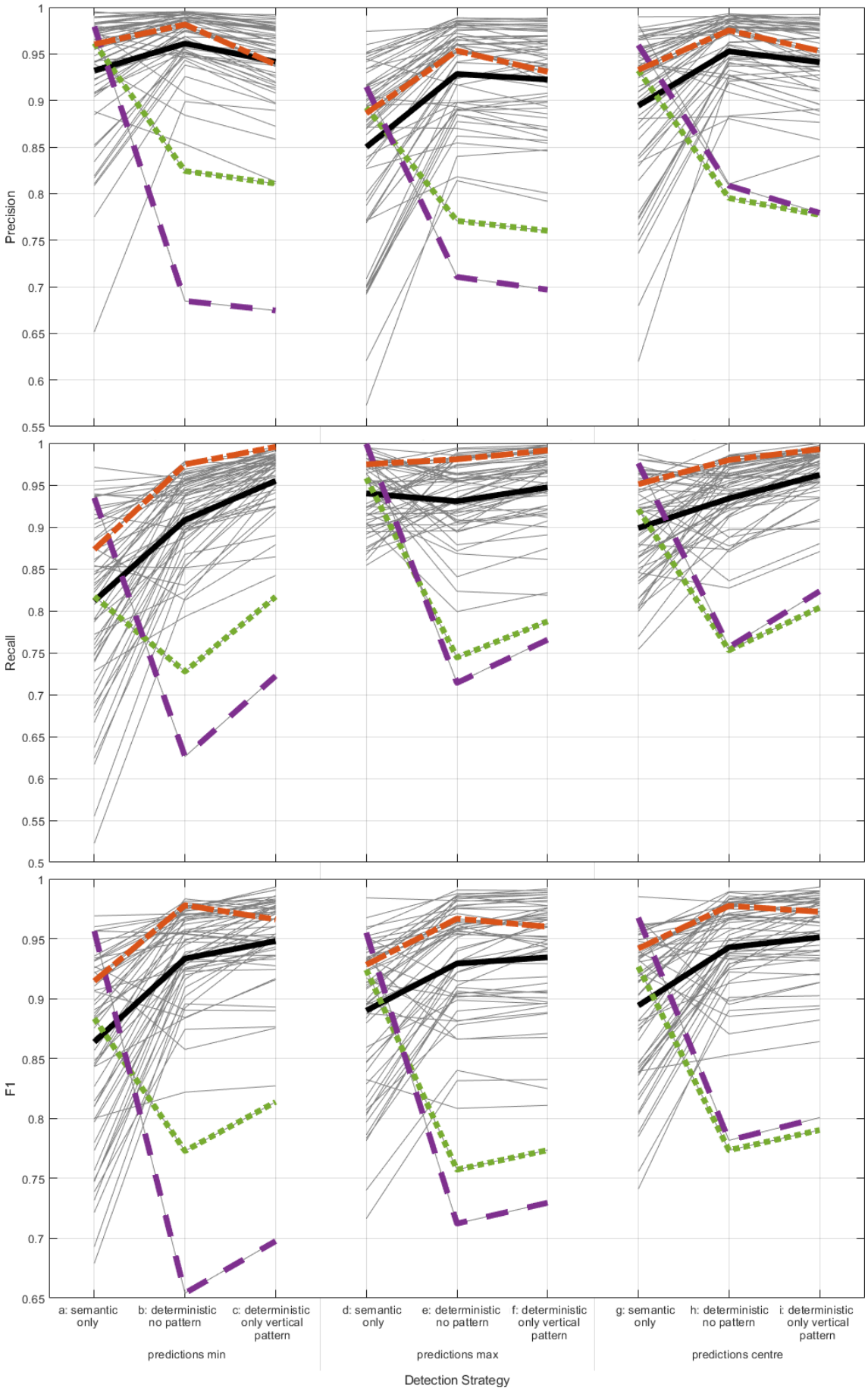
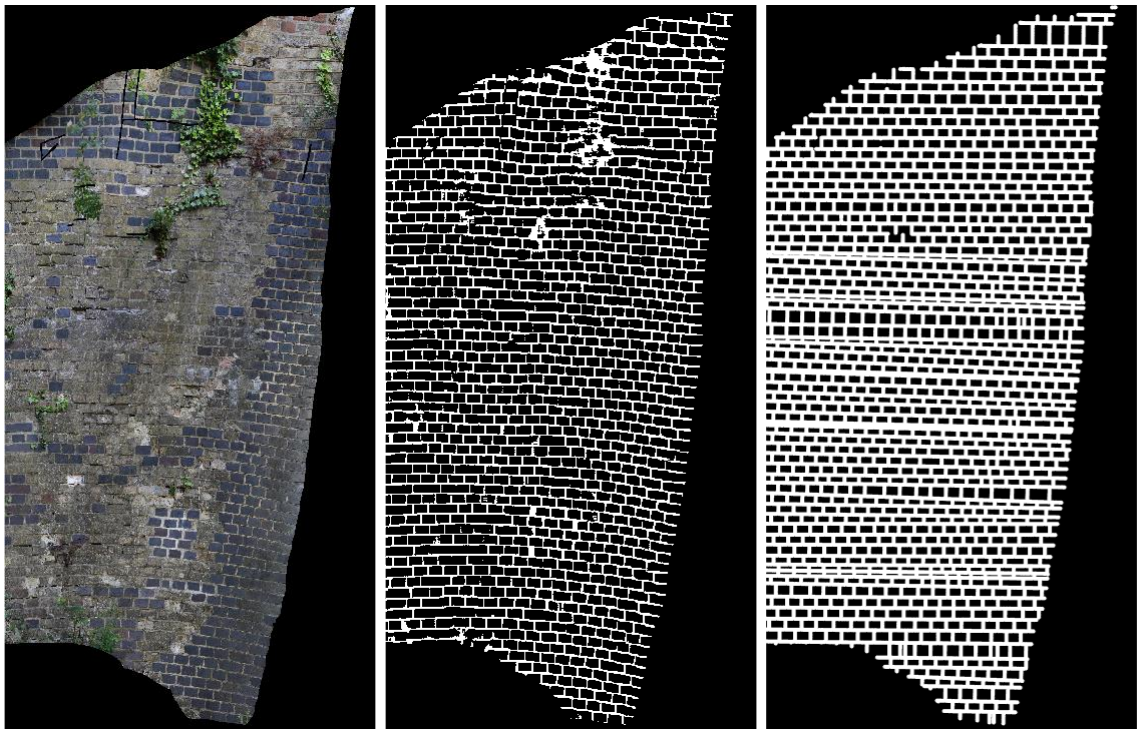
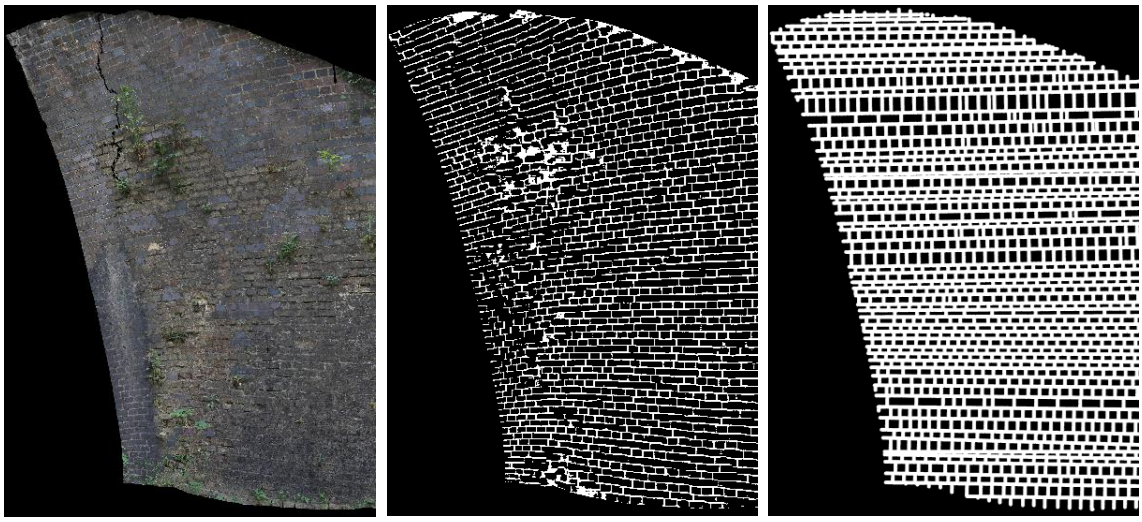


Figure 5-17: Performance metrics for individual images



a) b) c)

Figure 5-18: Output for image denoted by a green marking in Figure 5-17 showing: a) input image, b) centre mortar semantic segmentation output, c) deterministic mortar joint detection applied to semantic segmentation output



a) b) c)

Figure 5-19: Output for image denoted by a purple marking in Figure 5-17 showing: a) input image, b) centre mortar semantic segmentation output, c) deterministic mortar joint detection applied to semantic segmentation output

Although the deterministic methodology has been shown to improve the output of the semantic deep learning classifier, situations like this show the disadvantage in such a deterministic methodology, and how its generalisability will never match a data driven approach. The semantic deep learning classifier was shown to greatly outperform the edge detection mask for categorising mortar joints, and as such improved mortar joint detection performance would be best achieved through improving such a deep learning classifier, without using a deterministic methodology.

This would be achievable through a larger training data set, from different bridges, increasing the exposure of the classifier to more masonry surfaces. Additionally, different patch sizes could be experimented with, as well as different semantic models to improve the training of the classifier such that it learns to better recognise the mortar joint pattern as well as the local image textures at mortar joints.

5.3 Chapter summary

The focus of this chapter has been detecting mortar joints in images of masonry, as it was found that none of the existing mortar joint detection techniques in the literature have been developed to cope with images with a level of noise and distractions that typically exist on masonry arch bridges. Therefore, approaches to detect mortar in masonry images have been explored where the mortar has been covered, for example by paint, dirt, or vegetation. The chapter has outlined two different approaches to mortar joint detection, a deterministic approach, and a machine learning approach.

The deterministic methodology for detecting patterns in mortar joints was able to accurately determine mortar line locations in images with a resolution varying from 0.15 to 3.02 pixels per mm. The methodology was tested on images framing varied sizes of masonry wall and was found to perform most accurately for an image size of approximately 15 stretcher bricks wide. This gave the best balance between over-fitting a mortar line across too many bricks and the potential impact of noise in smaller images. The methodology was also tested successfully on masonry images from other bridges which showed differences in the appearance of the masonry. The deterministic methodology uses an edge detector to give an indication of mortar joint locations. Despite the edge detector giving a noisy output, the methodology is still able to determine locations of mortar joints.

Deep learning techniques for detecting mortar joints have then been explored. Firstly, an image window approach was explored, and this showed that as the sample window size

was increased, such that more masonry was imaged, the performance of the classification improved. However, this also resulted in an increase in the coarseness of classification, meaning such a classification is less useful. It was suggested that in the smaller image window sizes, classification was just based on localised texture, whereas a larger window size meant classification could be based upon interfaces and mortar pattern information.

This learning was applied by using a semantic deep learning classifier where image patches several brick courses in size were trained and tested, meaning local texture, local interfaces and global pattern information of mortar joints could be learnt. The output showed excellent performance, showing the classifier was able to accurately map mortar joints across almost the entirety of the noisy masonry surfaces tested. This was also shown to significantly outperform the edge detection output for detecting mortar joints, and therefore a trial was attempted where this semantically segmented mask was used as an input to the developed deterministic methodology. This trial was based on the assumption that even though this semantic classifier is more sophisticated and robust than edge detection, it still can mistake the positioning of mortar lines where the image conditions deteriorate. It has been shown that the deterministic pattern recognition methodology is able to enhance the output of a sophisticated deep learning based semantic mortar joint detector by enhancing detection in noisy regions of images using the pattern determined in clearer regions.

However, there are still situations where the deterministic methodology fails to accurately detect mortar where the mortar in the image is irregular. This therefore suggests that, although the deep learning based semantic mortar joint detector already works quite well, the best way to improve mortar joint detection further is through improving the training of a deep learning semantic classifier. A route to achieve this is through framing larger areas of masonry in the training data. This could aid the learning of the global mortar pattern, so the classifier is relying less on the potentially noisy local texture for classification. This way, the classifier would still be able to overlook instances of distorted mortar patterns that are the pitfall of the deterministic methodology.

6 EFFECT OF MORTAR JOINTS ON THE PERFORMANCE OF DEFECT CLASSIFICATION

Masonry surfaces are inherently noisy due to the mortar joints. In the literature review in Chapter 2.4 it was reported that previous studies have found that mortar joints are often falsely detected as defects, and previous researchers have suggested that segmenting mortar joints from masonry surfaces could lead to a more accurate detection. For masonry arch bridges, which are often old and remote, other distractors are also often present such as dirt, paint, and vegetation. These distractors further compound the problem when compared to clean masonry. The effect of mortar joints on the defect classification performance is therefore quantified in this chapter.

Figure 6-2 shows the results of a defect detection technique that has been developed as part of this work. This is based on a simple Canny edge detection methodology, which is far more rudimentary and less capable of differentiating defects from noise than deep learning methods. It is obvious in the edge detection output shown in Figure 6-2a, that it would be impossible to detect defects through edge detection directly, as the mortar joints are far more prominent than any defects present. Therefore, the developed edge detection based defect detection technique initially detects and masks mortar joints in the image, using the edge detection based deterministic methodology described in Chapter 5.1. Edge detection is then applied to the masked image, such that all mortar joint interfaces have been removed from the edge detection output.

Further refinement of the edge detection output is then performed to reduce the noise that is still detected. This refinement has been broken down in Figure 6-1. In Figure 6-1b, a Hough transform is used to detect small line segments in the edge detection mask. These line segments are grouped iteratively with those adjacent to themselves (as in Figure 6-1c) and each group is then grouped with other groups of line segments using a search area based on the predominant orientation of the lines in each group (as in Figure 6-1d). This grouping is performed because defects are typically either focused in a single localised

region, for example spalling, or have a long and thin structure, for example cracking. The output of grouping is shown in Figure 6-1e, upon which a threshold is used, such that only those groups of lines containing a certain number of points are retained, therefore removing the smaller groups that are most likely to be noise.

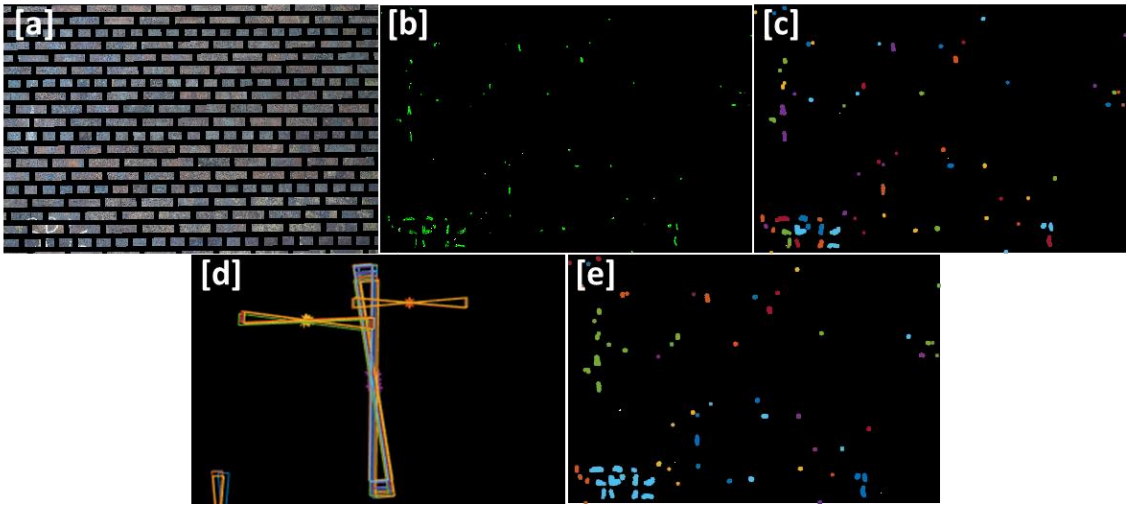
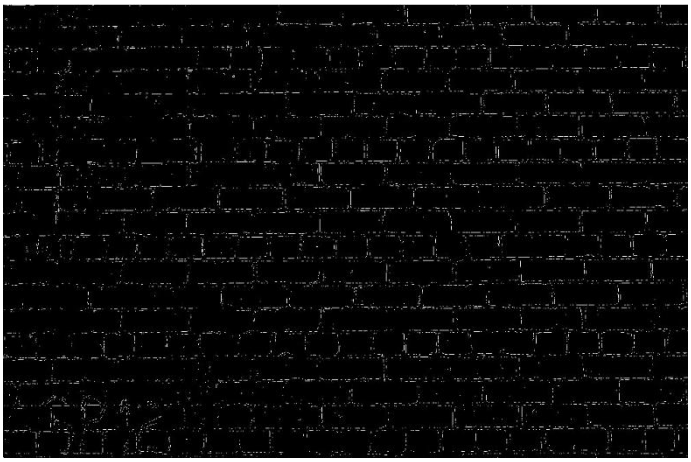


Figure 6-1: Edge detection based defect detection methodology: a) mortar joints masked, b) straight lines detected in a Sobel output, c) detected lines grouped with surrounding lines, d) search regions for line groupings based on orientation of lines in group, e) straight line groups after iterative grouping

The resulting defect map for the same image is shown in Figure 6-2b. In three locations, highlighted in green, defects were identified successfully. In four other locations, anomalies were correctly identified, but were not due to defects (or damage) of the bridge. In one of these locations, writing on the brick surface was accurately detected. The other three location all involve instances where mortar lines were not fully masked and therefore were detected. These masking errors were caused by either a thicker than normal mortar joint or an irregular brick spacing pattern.

In this case filtering out mortar lines from masonry images has meant that the noise in the edge detection output has been reduced such that defects are discernible and could be automatically separated, which would not otherwise have been possible. However, there are more robust classifiers than edge detection, as demonstrated in Chapter 5 for mortar joint detection, which are far better able to differentiate between mortar interfaces and other edge-like defects. The focus of this chapter is testing one such deep learning based classifier, to determine the effect mortar joints have on its performance.

a)



b)

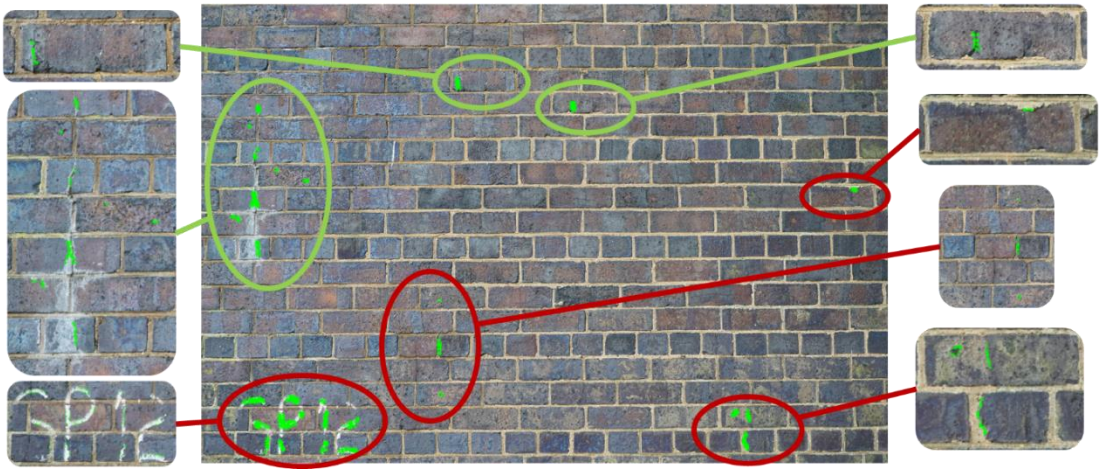


Figure 6-2: Edge detection based defect detection using a mortar line mask: a) base edge detection output, b) defect detection output having masked mortar regions and filtered results

6.1 Deep learning based defect detection

6.1.1 Methodology

This work uses a classification approach based on image windows. Images generated in the datasets developed in Chapter 4 have been segmented into smaller image patches, each 100 pixels in size, using a sliding window technique with a stride of 25 pixels between each image window. The 100-pixel image size, relative to the 155 pixel brick height in the images in the dataset, ensures that some image windows contain purely brick regions, while others contain a mixture of brick and mortar regions. Each image window is assigned a class based on the annotations of the pixels it contains. The central third of the image window is used to assign the class – if any pixels are annotated as defective in this region, then the image window is classified as such. Example image window patches for the different defect classes are shown in Figure 6-3. In this work, all the different

defective classes have been grouped into a single defective class, such that a binary classification is performed, with the image windows being classified either as clean or defective.

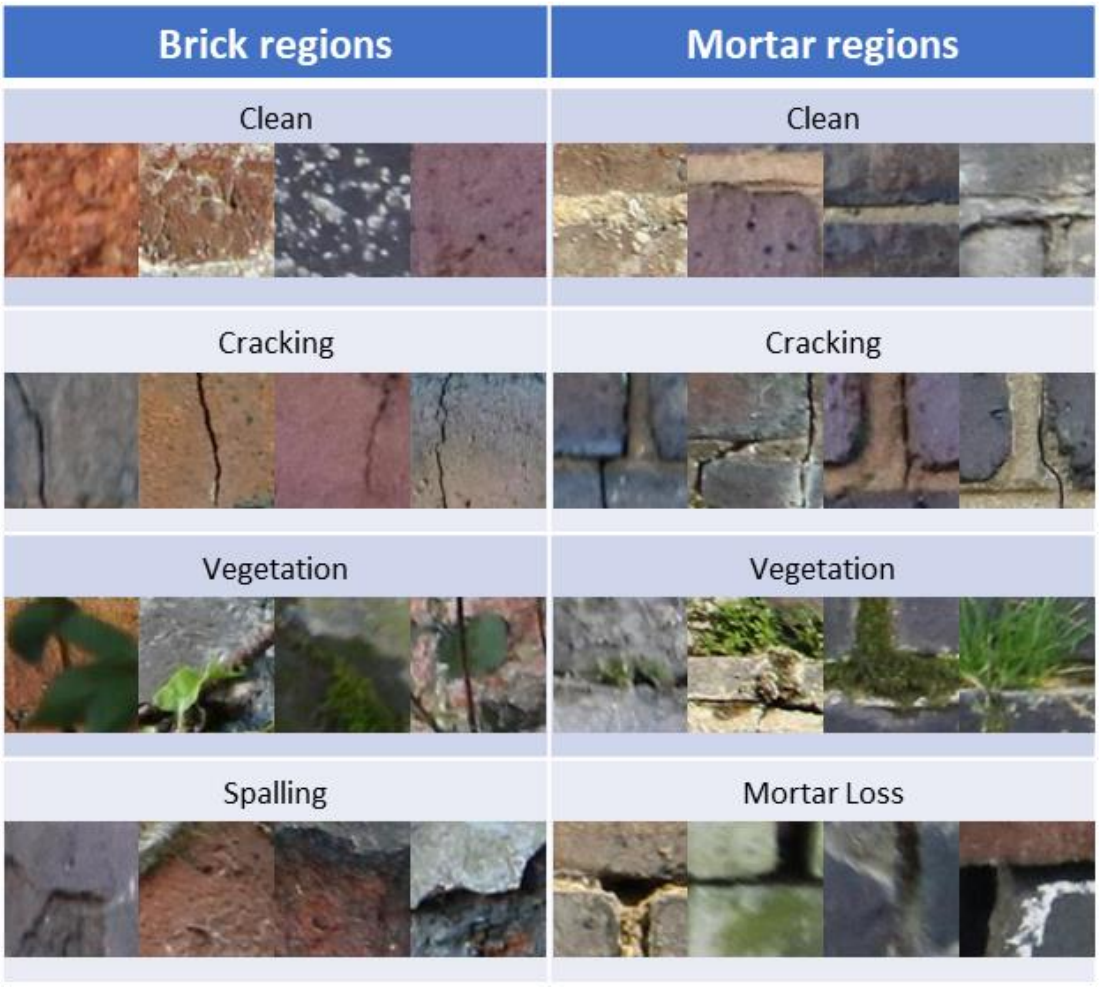


Figure 6-3: Example image window patches for different defect classes

The generated image windows are used to train and test a classifiers ability to learn the different defect classes. As before, a CNN based classifier has been used due to the suitability of the architecture for image classification tasks. As in Chapter 5.2.1, the GoogleNet Inception v3 architecture (Szegedy et al., 2016) is used. This architecture is shown in Figure 6-4. With this architecture, several convolution layers (depicted in yellow) are performed in parallel as part of each inception block. This enables the network to utilise convolutional filters of multiple sizes on the same level, such that both more globally, and more locally distributed features can be realised. This therefore means that defects of varied sizes and prominence within each image window can be trained to be recognised.

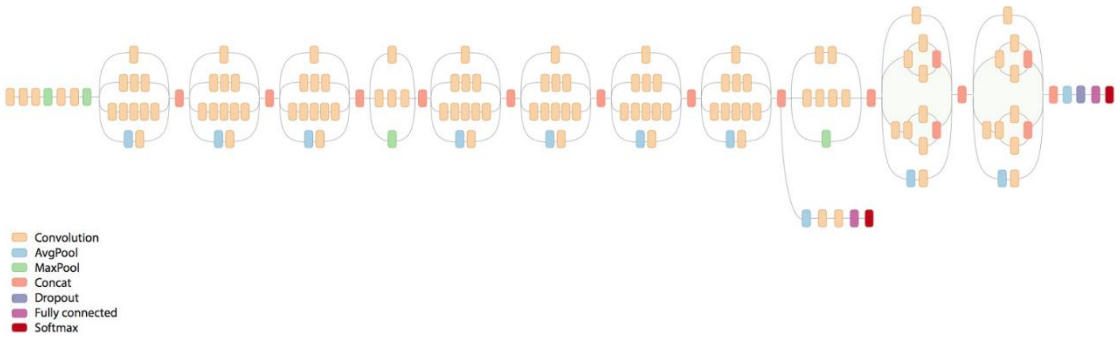


Figure 6-4: GoogleNet Inception v3 architecture (Szegedy et al., 2016)

Transfer learning has been used to pre-train the model as this has been shown to increase the generalisability of the trained model, as well as potentially reducing the size of the training set required to adequately train the model. The model has been pre-trained on the ImageNet dataset (Deng et al., 2009) which contains 1.4 million individual images divided into 1000 classes. In ImageNet, these classes are generally object-based classes such as ‘building’ or ‘ball’ and so are quite distant from the ‘defect’ / ‘clean’ classes being retrained to. Therefore, the features learnt during pre-training may not be so useful for classifying on this new domain. This is particularly the case at the higher layers closer to the output which filter more specific features of images.

Due to this, transfer learning has been performed using a fine-tuning approach, whereby a greater proportion of the network is retrained than in the feature extraction transfer learning approach used in Chapter 5.2.1, where only the top classification layer was retrained to the new classes. This involves a two-step process. Firstly, the final classification layer of the network (the red Softmax layer in Figure 6-4) is removed and replaced with a new randomly initialised classification layer for the new classification classes. This is then trained with the rest of the weights in the network fixed. This is akin to the training performed in Chapter 5.2.1. Secondly, other weights in the deeper layers of the network are unfixed and made trainable, and the network is then trained again to fine-tune the weights for these layers, using a much smaller learning rate than previously employed, such that the weights are fine-tuned from their previous optimisation. Training is performed in this two-step process, such that the weights of the deeper layers of the network are not destroyed by the randomly initialised top classification layer for the new classification classes.

Training has been performed using the Keras machine learning framework (Chollet and Others, 2015), with the Tensorflow backend (Abadi et al., 2016). The Inception v3 model uses batch normalisation after convolution layers to stabilise the learning process in deep

models. At the time this work was conducted, the default Keras implementation of batch normalisation with transfer learning meant that a significant performance degradation was noticed between the testing and training phases, especially in cases where the original dataset is considerably different to the retrained one. This is due to the batch normalisation statistics of layers that were frozen during training being implemented differently during testing from training, using statistics of the training batches during training and the original dataset during testing. For this work, a custom implementation of Keras, which corrects this implementation of batch normalisation of frozen layers in training, has therefore been used (Vryniotis, 2018).

For training the network, a hyperparameter search has been performed, to maximise accuracy on validation data. The results of this search have been presented in Appendix 2. Concluding from this search, the hyperparameters used are as follows. During top layer training, a fully connected dense layer size of 1024 neurons and a Softmax layer with 2 classes are trained using a Root Mean Squared Propagation (RMSProp) optimiser with a learning rate of 0.001, trained across 50 epochs of the training data with a batch size of 50 images. During fine-tuning training, zero layers are frozen, so all are trainable. These are trained with a Stochastic Gradient Descent (SGD) optimiser with a learning rate of 0.0001, 0.9 momentum and no decay, across 50 epochs of the training data with a batch size of 50 images. Image augmentation in the form of shearing, rescaling, and flipping is used in training.

6.1.1.1 Defect detection strategies

To determine the effect of mortar joints on defect detection accuracy, four different classification strategies have been tested. These are summarised in Figure 6-5. From left to right, the first strategy does not use any mortar joint information at all. In this strategy, only the classes of ‘defect’ and ‘clean’ surfaces are trained, with image window patches showing either mortar regions or brick regions. The second strategy defines separate defect classes for mortar and brick regions, so that there are two defect classes for each defect type, one for the defect occurring in mortar and one for the defect occurring in brick. The third strategy completely separates the mortar and brick regions and uses a separately trained classifier for image patches in each region. The fourth and final strategy goes one step further by blacking out the non-mortar, or non-brick regions in each mortar or brick image window, respectively. This is done through zero padding the image, replacing all the blacked-out pixel values with zeros, such that they do not contribute to the classification.

In both the third and fourth classification strategies, the different sets of classified images are merged so that the image windows being classified are the same for all three detection methodologies. Additionally, for all the defect classification methodologies, only the image window patches that show fully brick regions or are centred on mortar regions are examined. Therefore, those image window patches that partly contain both brick and mortar regions are removed. As there is an overlap between image window patches, the whole of the masonry surface is still included in at least one examined patch. This step has been taken to ensure consistency of the data being examined by the four classification methodologies.

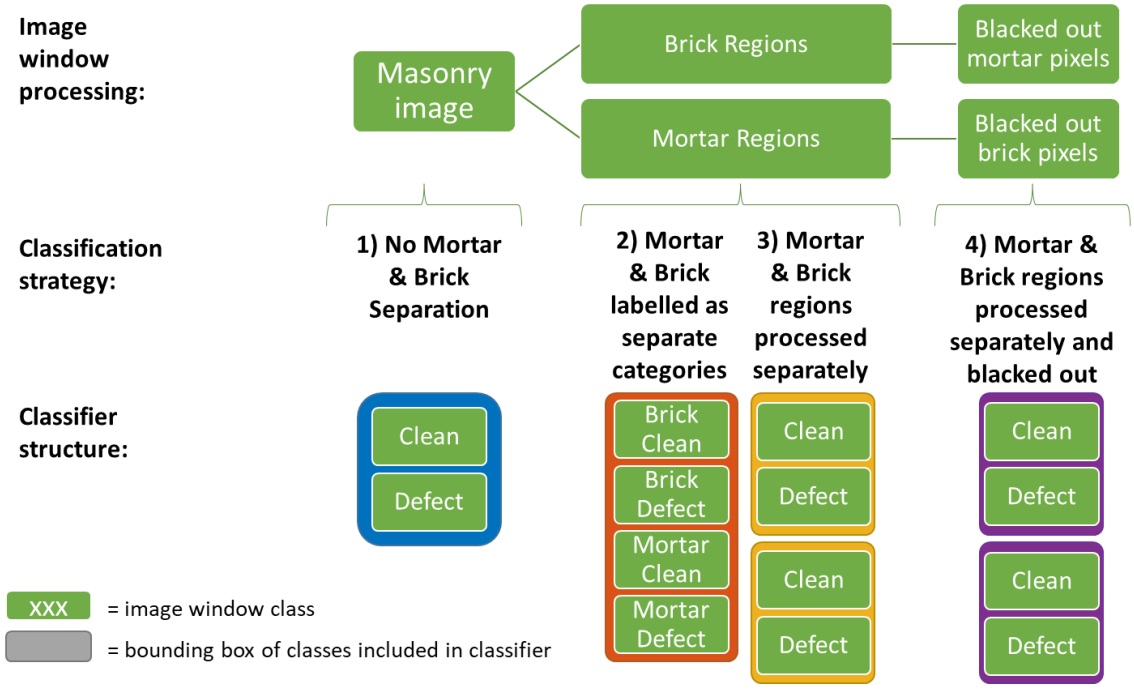


Figure 6-5: Different classification strategies for investigating the effect of mortar joints on the accuracy of defect detection

As stated above, this work uses annotated images from the datasets created in Chapter 4. The training dataset comprises the data of bridges which are primarily viaducts denoted by blue markers in the map in Figure 4-2. From this dataset, 94 key areas of the bridges where defects are prevalent have been focused on, and these images were manually orthogonalized to the brick surface. Nevertheless, undamaged masonry surfaces are much more prevalent in the dataset than defective surfaces, so to achieve a balanced dataset between the classes for training, image windows are randomly discarded. The resulting training dataset for each class consists of 24,000 image windows, including 5000 used for validation.

The testing dataset comprises the data of the three bridges denoted by red markers in the map in Figure 4-2. For this dataset, photogrammetry and UV mapping have been used to create orthogonalized image textures for the entire bridge surface, as discussed in Chapter 4.5. Therefore, the training and testing data has been taken from completely separate sets of bridges, testing the ability of the different methodologies to generalise. The resulting testing dataset consists of 1,846,601 image windows. The class distribution of these image windows is the same as that seen on the bridge surfaces, to test the performance of the different classifiers in conditions that would be seen in practice. As a result, there is an imbalance between the defective and clean classes, with 323,537 defective and 1,523,064 clean image windows.

Due to the orthogonalization process, only flat areas of bridges, such as piers and spandrels have been included in the dataset used for training, whereas the testing dataset includes all surfaces of the bridge, including the curved arch barrel. This is because UV mapping enables the unravelling of an arch barrel without scaling parts of the image, as demonstrated in Chapter 4.5.1.2. However, this does mean that the methods are tested on parts of bridges that were not included in the training set, where surfaces and defects may appear different. Therefore, an additional test has been performed, in which the different methodologies were trained on a subset of 85 of the images in the previous training dataset and tested on the remaining 9 images of the data. For this test, the training and testing data therefore contained images from the same bridges, though separate parts of the bridges, still testing generalisability. In this case, 23,500 image windows were used for training for each class, of which 4,500 image windows were used for validation. The methodologies were then tested on 64,404 image windows, of which 7,460 were defective and 56,944 were clean.

6.1.1.2 Metrics used to compare different classifier performances

The confusion matrix (Table 6-1) summarises the output of the classification, comparing the predicted classification of image windows against their true classifications. The two correct classification possibilities are true positive and true negative, shown in green in the top left and bottom right corners, respectively. A true positive classification represents an image window which both contains a defect and has been predicted to be defective by the classifier. Conversely, a true negative classification represents an image window which both contains no defects and is predicted to have no defects by the classifier. The two incorrect classification possibilities are false negative and false positive, shown in red in the top right and bottom left corners, respectively. A false negative classification

represents an image window which contains a defect but has been predicted to have no defects by the classifier, and a false positive classification represents an image window which contains no defects but is predicted to contain a defect by the classifier. Therefore, a good classifier will have high numbers of correctly classified windows, and few incorrectly classified windows.

The different performance metrics in Table 6-2 summarise the data in the confusion matrix - the different metrics prioritise different attributes. The calculated metrics are as follows:

Specificity: The true negative rate, i.e., the proportion of non-defective image windows that are classified as non-defective.

$$Specificity = \frac{number\ of\ true\ negatives}{number\ of\ true\ negatives\ +\ number\ of\ false\ positives} \quad (6\text{-}1)$$

Sensitivity/Recall: The true positive rate, i.e., the proportion of defective image windows that are classified as defective.

$$Sensitivity/Recall = \frac{number\ of\ true\ positives}{number\ of\ true\ positives\ +\ number\ of\ false\ negatives} \quad (6\text{-}2)$$

Precision: The proportion of the image windows that were classified as defective that in fact were defective.

$$Precision = \frac{number\ of\ true\ positives}{number\ of\ true\ positives\ +\ number\ of\ false\ positives} \quad (6\text{-}3)$$

F1 score: The harmonic mean of precision and recall.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (6\text{-}4)$$

Balanced accuracy: The average of specificity and sensitivity. A measure of accuracy for imbalanced datasets.

$$Balanced\ accuracy = \frac{Specificity + Sensitivity}{2} \quad (6\text{-}5)$$

The different metrics of balanced accuracy and F1 score both function as measures of the accuracy of the classifier. Both have been included, as the F1 score focuses on the positive class, i.e., the windows that are classified as defective, and is not affected by the number of true negatives, i.e., the number of non-defective image windows that have been

correctly classified as such. Focusing on the positive class is important, as the ability to accurately discern defects is key to a classifier. However, because the dataset is imbalanced, with far more non-defective (negative class) image windows than defective, the metric of balanced accuracy is also important to establish a classifiers performance since it focuses on the ability to detect both the positive (defective) and negative (non-defective) classes.

All the above metrics do not score how confident the classifier is on a particular decision. For instance, a classification with a confidence of 0.6 would be judged the same as one with a confidence of 0.9. As a result, in Table 6-2 the Brier score has also been calculated, and receiver operator characteristic (ROC) curves have been plotted in Figure 6-7 to Figure 6-13.

The Brier score is a proper scoring rule equal to the mean square error between the predicted and true probabilities. It is calculated as:

$$Brier\ Score = \frac{1}{\mathrm{N}}\sum_{t=1}^{N}(f_t - o_t)^2 \tag{6-6}$$

where f_t is the predicted probability and o_t is the actual probability of image window t being defective, and N is the number of image windows in the dataset. The actual probability of an image window containing a defect is 1 where an image window contains a defect, and 0 when it does not. The Brier score, in this formulation, is between 1 (the worst score achievable) and 0 (the best score achievable).

An ROC curve plots the true positive rate (sensitivity) on the y-axis against the false positive rate (1 - specificity) on the x-axis, at different threshold values. The true positive rate represents the probability of detecting a defective window, and the false positive rate represents the probability of misclassifying a non-defective window as defective. Therefore, increasing the classification threshold would cause a decrease in both the true positive rate and the false positive rate, as fewer defective windows would be classified, and fewer non-defective windows would be misclassified. A truly random classifier would therefore give a point along a diagonal line from (0,0) to (1,1). A perfect classifier would give the point (0,1), where true positive rate equals 1, and false positive rate equals 0. Therefore, the better performing a classifier, the closer to the (0,1) point its curve approaches, and the larger the area under its curve.

6.1.2 Results

6.1.2.1 Results from testing on dataset of three independent bridges

Firstly, the results from testing on the dataset of the three complete bridges denoted by red markers in the map in Figure 4-2 are examined.

Figure 6-6 visually shows the output of the classifier on different bridge surfaces within this dataset. For simplicity, only the output of the classifier using classification strategy 1 (no mortar and brick separation) is shown as the different classification strategies visually give a similar output as demonstrated in Figure 6-15. The two masonry surfaces depicted in Figure 6-6 show vastly differing image conditions.

The left images show a section of the SBR12 bridge spandrel wall where the masonry is in direct sunlight such that some mortar joints exhibit shadowing. This is not too extreme as the mortar channel is shallow. There is also a shadow interface at the top of the image caused by a step in the masonry at the base of the parapet wall. The classifier was mostly able to ignore these shadows to give a good agreement with the ground truth classifications.

The right images show a section of the SBR8 bridge arch barrel. On this bridge the arch barrel suffers from thick soot like deposits making interpretation difficult. The image depicts a region of the arch barrel where there is a major diagonal fracture as well as several smaller defective regions. The classifier has been able to detect the major fracture in its entirety, as well as most of the smaller defects. It has also mostly avoided falsely classifying non-defective surfaces where the soot markings have the potential to look defective.

Next, the quantitative classification performance across the whole dataset for the different classification strategies is examined. This is presented in the confusion matrix in Table 6-1, with performance metrics summarised in Table 6-2. These tables present the outputs for both the four different classification strategies (labelled 1 to 4) summarised in Figure 6-5 above, as well as seven different data processing strategies (labelled a to g) that will be discussed in due course. These classification strategies and data processing strategies are also summarised in the footnotes of Table 6-1 and Table 6-2.

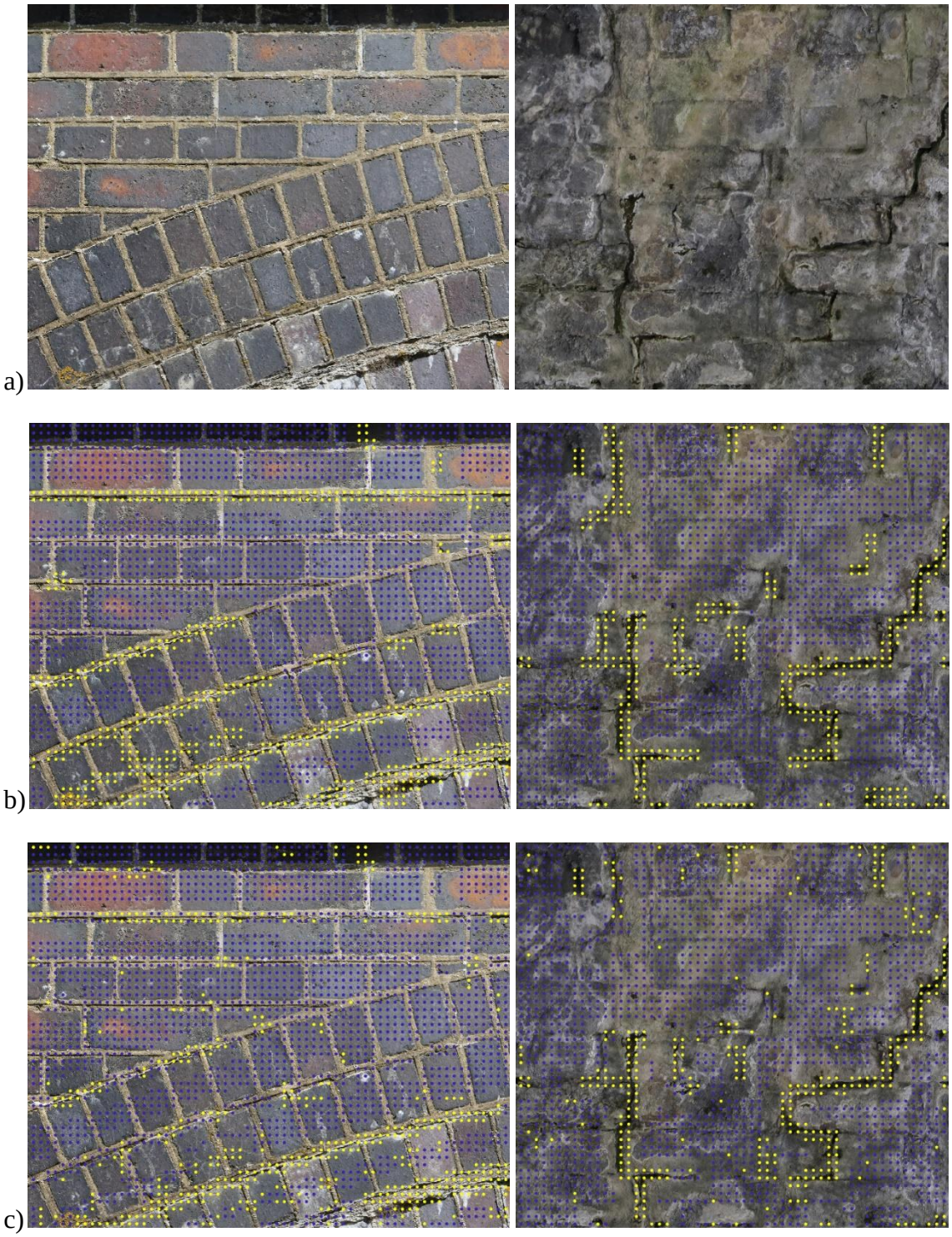


Figure 6-6: Closeup images showing output of automated defect classifier for classification strategy 1, where no mortar and brick separation is performed, on bridge surface SBR12 spandrel wall (left) and SBR8 arch barrel (right): a) raw image, b) ground truth, c) automated defect classification predicted with 80% certainty

Table 6-1: Confusion matrix comparing the performance of the different classification strategies for defect detection in masonry

				Classification strategies							
				1	2	3	4	1	2	3	4
				Predictions							
				Defective				Clean			
Data strategies	a	Actual	Defective	208,809	197,261	193,671	200,892	114,728	126,276	129,866	122,645
	b			128,058	117,131	113,949	123,941	108,144	119,071	122,253	112,261
	c			42,795	43,822	40,887	45,517	24,542	23,515	26,450	21,820
	d			166,014	153,439	152,784	155,375	90,186	10,2761	103,416	100,825
	e			240,522	227,986	223,106	236,257	83,015	95,551	100,431	87,280
	f			154,382	142,245	136,878	150,560	81,820	93,957	99,324	85,642
	g			120,784	117,562	117,563	121,345	11,302	14,524	14,523	10,741
	a		Clean	205,693	227,997	179,869	252,832	1,317,371	1,295,067	1,343,195	1,270,232
	b			205,693	227,997	179,869	252,832	1,317,371	1,295,067	1,343,195	1,270,232
	c			37,612	55,542	35,532	49,205	507,154	489,224	509,234	495,561
	d			168,081	172,455	144,337	203,627	81,217	805,843	833,961	774,671
	e			205,693	227,997	179,869	252,832	1,317,371	1,295,067	1,343,195	1,270,232
	f			205,693	227,997	179,869	252,832	1,317,371	1,295,067	1,343,195	1,270,232
	g			205,693	227,997	179,869	252,832	1,317,371	1,295,067	1,343,195	1,270,232

Classification strategies: (1) no mortar & brick separation, (2) mortar & brick labelled as separate categories, (3) mortar & brick regions processed separately, (4) mortar & brick regions processed separately and blacked-out.

Data strategies: (a) Raw data, (b) excluding vegetation regions, (c) only brick regions, (d) only mortar regions, (e) assume defect is identified if more than 50% is identified, (f) excluding vegetation regions and assume defect is identified if more than 50% is identified, (g) assume defect is identified if more than 50% is identified and defects less than 1/3 header brick in size (7500px) excluded.

Table 6-2: Performance metrics comparing the performance of the different classification strategies for defect detection in masonry

Performance metric	Data strategies	Classification strategies			
		1	2	3	4
Specificity	a	0.865	0.850	0.882	0.834
	b	0.865	0.850	0.882	0.834
	c	0.931	0.898	0.935	0.910
	d	0.828	0.824	0.853	0.792
	e	0.865	0.850	0.882	0.834
	f	0.865	0.850	0.882	0.834
	g	0.865	0.850	0.882	0.834
Sensitivity/recall	a	0.645	0.610	0.599	0.621
	b	0.542	0.496	0.482	0.525
	c	0.636	0.651	0.607	0.676
	d	0.648	0.599	0.596	0.607
	e	0.743	0.705	0.690	0.730
	f	0.654	0.602	0.580	0.637
	g	0.914	0.890	0.890	0.919
Precision	a	0.504	0.464	0.519	0.443
	b	0.384	0.339	0.388	0.329
	c	0.532	0.441	0.535	0.481
	d	0.497	0.471	0.514	0.433
	e	0.539	0.500	0.554	0.483
	f	0.429	0.384	0.432	0.373
	g	0.370	0.340	0.385	0.324
F1 score	a	0.566	0.527	0.556	0.517
	b	0.449	0.403	0.430	0.404
	c	0.579	0.526	0.569	0.562
	d	0.563	0.527	0.552	0.505
	e	0.625	0.585	0.614	0.582
	f	0.518	0.469	0.495	0.471
	g	0.527	0.492	0.547	0.479
Balanced accuracy	a	0.755	0.730	0.740	0.728
	b	0.704	0.673	0.682	0.679
	c	0.783	0.774	0.771	0.793
	d	0.738	0.711	0.724	0.699
	e	0.804	0.778	0.786	0.782
	f	0.759	0.726	0.731	0.736
	g	0.890	0.870	0.886	0.876
Brier score	a	0.131	0.150	0.133	0.158
	b	0.135	0.154	0.136	0.161
	c	0.079	0.102	0.082	0.092
	d	0.157	0.173	0.158	0.190
	e	0.116	0.134	0.117	0.140
	f	0.122	0.140	0.123	0.146
	g	0.099	0.112	0.091	0.122

Classification strategies: (1) no mortar & brick separation, (2) mortar & brick labelled as separate categories, (3) mortar & brick regions processed separately, (4) mortar & brick regions processed separately and blacked-out.

Data strategies: (a) Raw data, (b) excluding vegetation regions, (c) only brick regions, (d) only mortar regions, (e) assume defect is identified if more than 50% is identified, (f) excluding vegetation regions and assume defect is identified if more than 50% is identified, (g) assume defect is identified if more than 50% is identified and defects less than 1/3 header brick in size (7500px) excluded,

The data processing strategies will now be explained. Firstly, in data strategy ‘a’ the raw data is examined using the confusion matrix in Table 6-1 and the performance metrics in Table 6-2. The greatest number of true positive detections are detected using classification strategy 1 (with no mortar & brick separation). As a result, classification strategy 1 also gives the best sensitivity/recall performance, as the greatest number of defective image windows are classified as such. Conversely, classification strategy 3 (where mortar and brick regions are processed separately) gives the greatest number of true negative detections, where non-defective regions are correctly classified as such. This results in the best specificity and precision performance. When these metrics are combined in the F1 score and balanced accuracy, classification strategy 1 and classification strategy 3 both show the best performance. This result is mirrored with the Brier score, and the raw data ROC curve in Figure 6-7 denoted by the solid lines, where strategies 1 and 3 perform best.

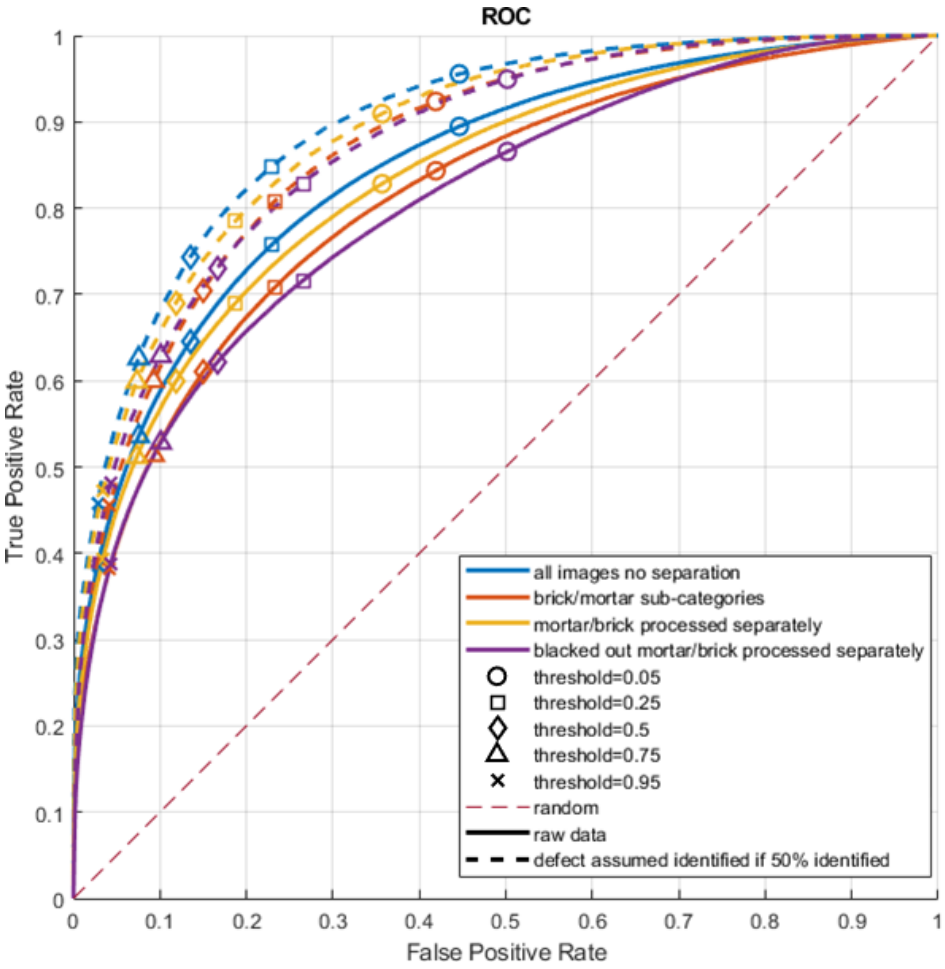


Figure 6-7: ROC plot comparing different strategies for defect detection in masonry

In Figure 6-7, different threshold levels giving rise to the respective true/false positive rates have also been plotted. It can be seen that at the 0.5 threshold, that is used in Table

6-1 and Table 6-2, classification strategy 3 (mortar and brick regions processed separately) has its threshold biased towards a lower true positive and false positive rate. That corroborates with the high specificity and precision performance found with this classifier described above, demonstrating the utility of the ROC curve compared to performance at just a single threshold.

For the F1 score, balanced accuracy, and Brier score, as well as on the ROC curve in Figure 6-7, classification strategy 4 (processing mortar and brick separately and blacking out respective contrasting region) has performed worst. This is possibly a result of the processing strategy employed. While classes are assigned to an image window based on the presence of a defect in the image window, that defect could be in a brick region for an image window focused on a mortar region, or vice versa. In the case of classification strategy 4, the defect would be blacked out. However, for comparative purposes, all 4 strategies have used the same ground truth data. To assess the impact of this, classification strategy 4 was re-analysed with its ground truth data based only on defects present in the regions that were not blacked-out. The results are shown in Figure 6-8. It is apparent that this results in an increase in performance for classification strategy 4 such that it performs similarly to the other strategies.

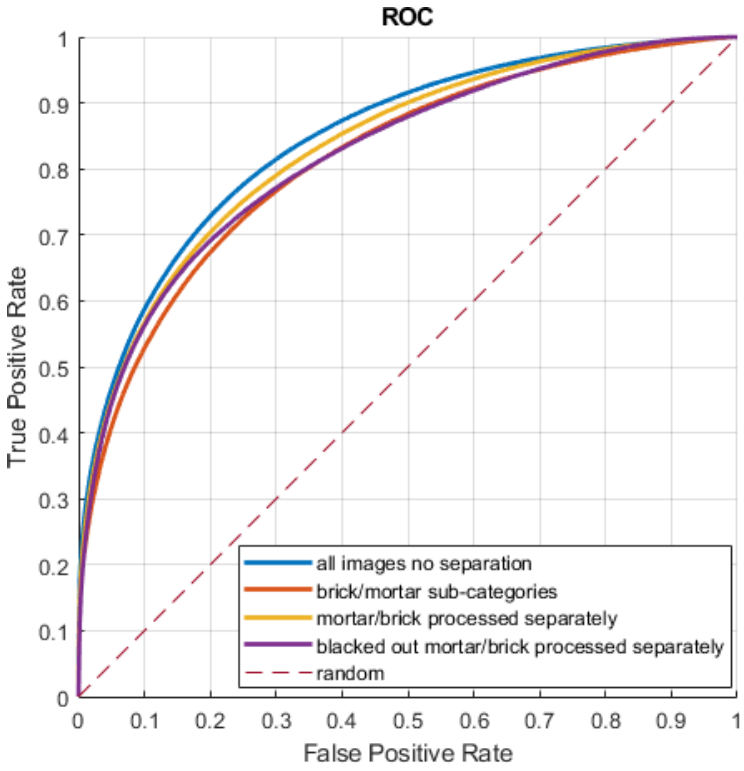


Figure 6-8: ROC plot comparing different strategies for defect detection in masonry with blacked-out classifications based on visible regions

Figure 6-9 and Figure 6-10 show the spread of performance across the different images tested. In Figure 6-9, this spread is across the different bridges tested, while in Figure 6-10 the spread is across the different component parts of those bridges, each being depicted by a separate image. From Figure 6-9, the performance of bridge SBR8 is better than that of SBR4, with SBR12 performing the worst. Referring to Chapter 4.2, SBR12 suffers from a layer of mud on the abutments from farm traffic, and SBR4 features a large proportion of surfaces that are painted, either through block painting or graffiti, with large areas of green painting partially mistaken for vegetation. SBR12 also heavily suffers from efflorescence and thick surface deposits on the arch barrel, which are classes of defect that have not been trained. All three bridges suffer from some images captured with intense sunlight resulting in shadowing, though this shadowing potentially effects a greater proportion of surfaces on SBR4 and SBR8 relative to SBR12. One effect of this is a shadow on the mortar interface where the mortar joint is recessed, which can easily be mistaken for a crack. From Figure 6-10, it is apparent that the spread in performance across the individual tested images is greater than the spread in performance across the classification strategies tested. This suggests that the conditions captured in the image have a bigger impact on classification performance than the classification strategy being tested.

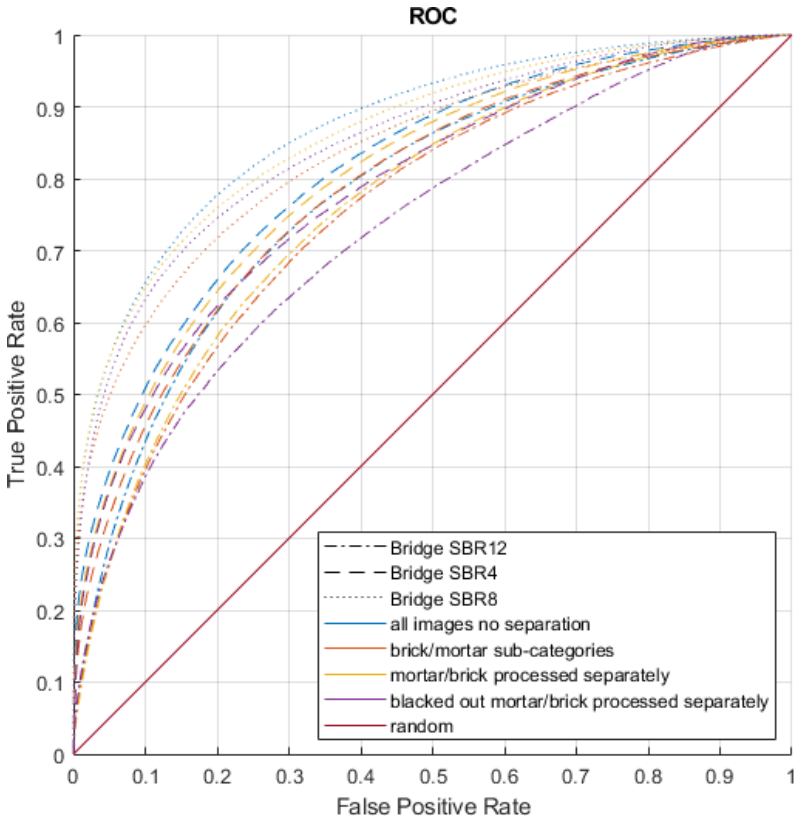


Figure 6-9: ROC plot comparing different strategies across individual bridges tested

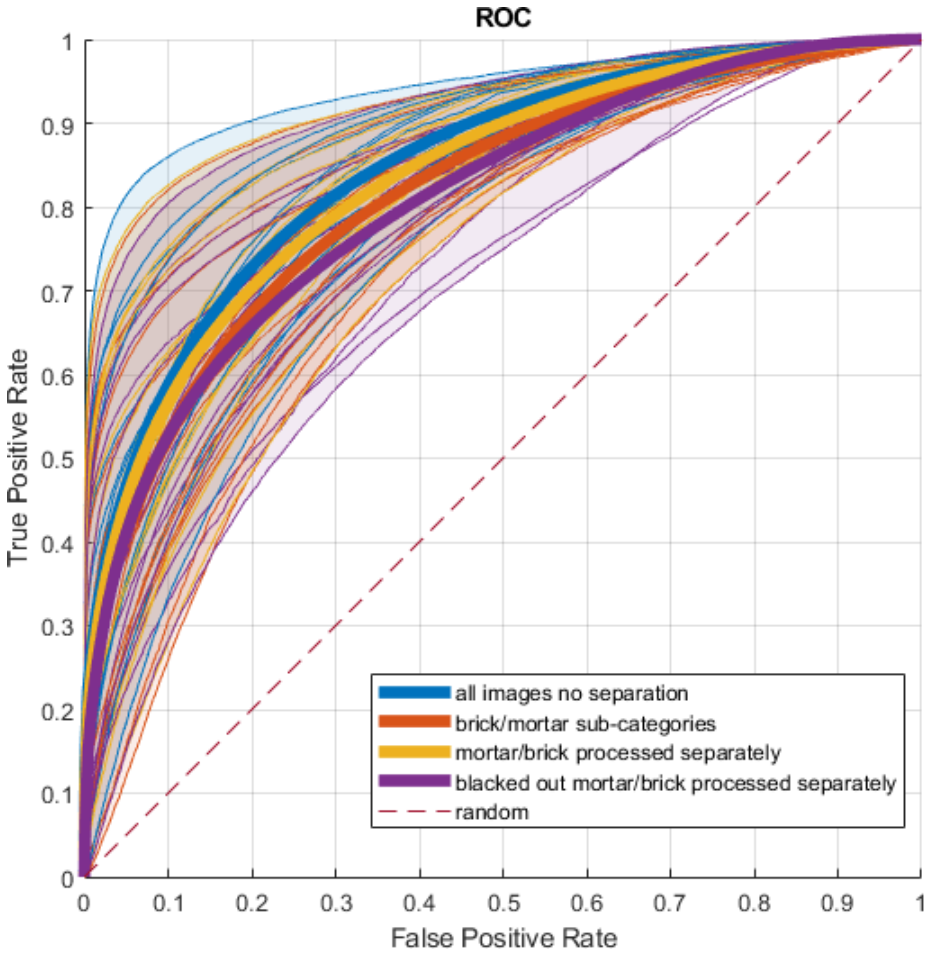


Figure 6-10: ROC plot comparing different strategies across the individual images tested

In vegetation regions, the vegetation mostly covers the mortar joint interface, and therefore different detection strategies to minimise adverse effects caused by the interface are likely to have insignificant effect. As a result, in Figure 6-11 the vegetation regions have been removed from the analysis. Referring to the raw data plot in this figure, denoted by solid lines, it is apparent that with vegetation regions removed, similar conclusions can be drawn as to the relative performance of the different classification strategies tested, though all strategies perform worse. This suggests that vegetation regions are generally predicted better than other defect types. This is also true for the confusion matrix in Table 6-1, and the performance metrics in Table 6-2 where data strategy ‘b’ shows the results with vegetation regions removed. Here the relative ranking of the different classification strategies remains the same as that without removing vegetation regions, for both the number of true positive and true negative classifications in the confusion matrix, and the different performance metrics.

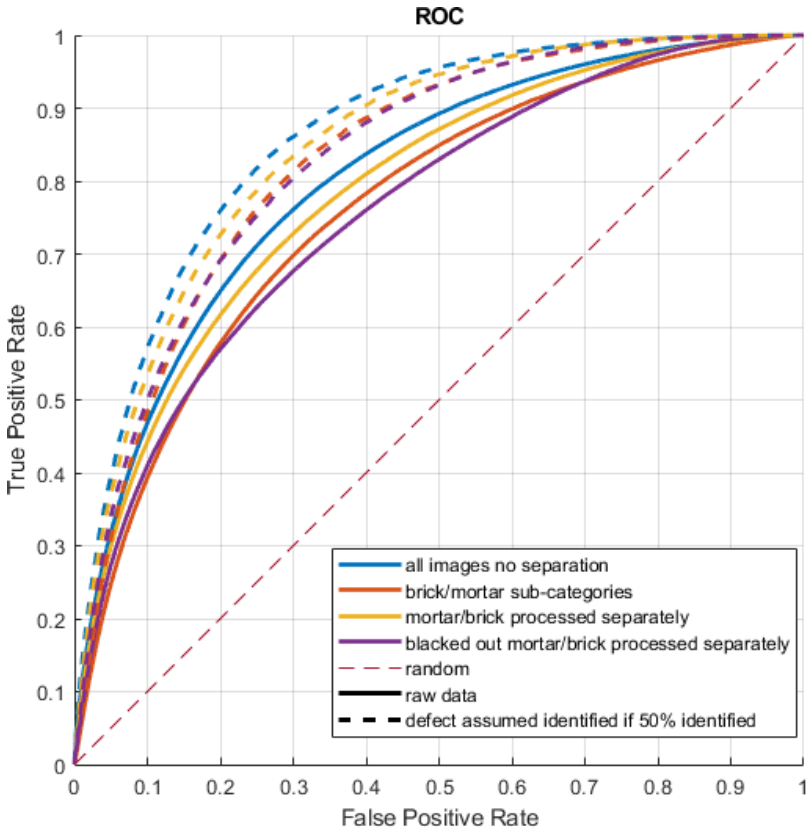


Figure 6-11: ROC plot with vegetation regions removed for different masonry detection strategies

The results examined so far all suggest that either the classification strategy 1, where all images are processed together with no mortar/brick separation performs best, or that there is not much difference in the performance of the different classification strategies, with features in the input image having far more of an effect on classification performance. This would lead to the conclusion that mortar joints do not influence the performance of classification significantly for this new classifier. In Figure 6-12 the performance of the mortar regions and the brick regions has been examined separately to test this. For all classification strategies, the classification performance in the brick only regions is superior to that in the mortar only regions. Higher performance in brick regions than mortar regions is also seen in Table 6-1 and Table 6-2, when comparing data strategies ‘c’ and ‘d’.

Additionally, with classification strategy 4, the blacked-out processing of mortar and brick regions separately, the relative performance in brick regions is better than that in mortar regions. This is to be expected because of the intrinsic blacking out of defects in image windows described above and demonstrated in Figure 6-8. A mortar image window is much more likely to contain brick pixels, and therefore a blacked-out defect, than a

brick image window is to contain mortar pixels. This is because of the narrow nature of mortar relative to brick, meaning mortar image windows face far more blacking-out.

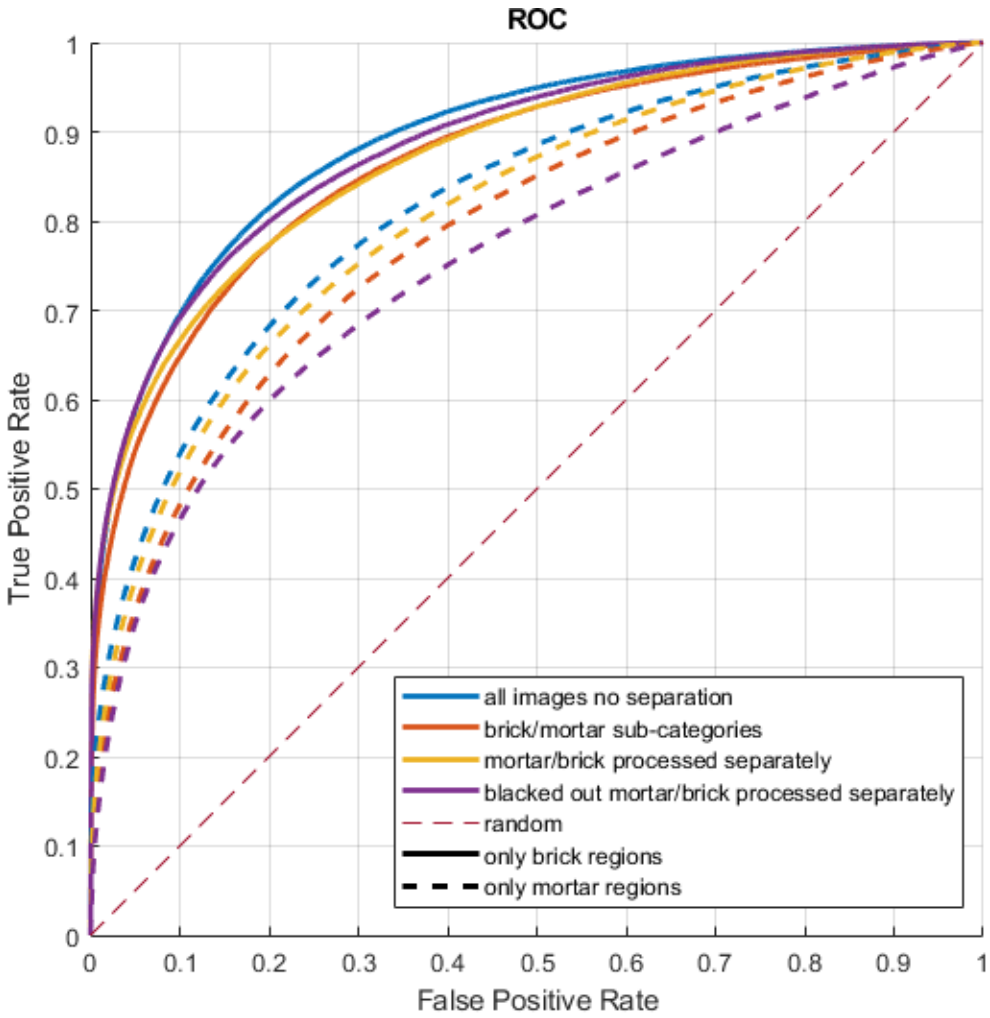


Figure 6-12: ROC plot comparing mortar and brick regions separately for different masonry detection strategies

If a defect has been mostly, but not fully identified, then an inspector processing the defects present on a bridge would find the utility of such a tool almost equal to as when a defect is fully identified. Additionally, as the defects have been manually annotated in the ground truth data, there is potential for incorrect classification of ground truth data. This would mean that not all of each ground truth defect is in fact a defect. As a result, a test has been conducted to augment the raw data from the output of each classifier such that if half of an individual isolated defect has been classified correctly, then the entirety of the isolated defect will be marked as classified correctly. This is fundamentally equivalent to saying that the defect was correctly detected, even if the precise extent, or size, of the defect was not correct. This has been done by calculating the median of the predicted probability of being defective for all the windows in each individual defect. Then the

prediction of all image windows within the defect with a prediction below the median is updated to the median. In this way, the ROC curve, which plots performance at different threshold levels, will recognise that the whole defect has been detected once the threshold level is above the median predicted value for the defect, i.e., the threshold value at which half of the defect would have been classified originally. The effect of this processing strategy is shown in Table 6-1 and Table 6-2 for data strategy ‘e’ for all image windows and ‘f’ for windows excluding vegetation regions. Here an improvement in the recognition of defective image windows is apparent, though as expected no change in the recognition of clean image windows. This translates to an increase in all performance metrics except specificity which focuses on the clean image window recognition. The relative performance of the different classification strategies remains unchanged. This behaviour is also examined in ROC plots in Figure 6-7 and Figure 6-11. These show an increase in performance across all classification strategies, though the relative performance remains unchanged.

A final data augmentation strategy identifies the effect of the size of defect on classification. This analysis has been performed as it is noted that larger defects are more likely to affect the structural performance of a bridge. Firstly, the effect of thresholding defect size is investigated: only defects greater than one third of the area of a header brick (7500px) in size are retained. The effect of this on the ground truth data for one wing-wall is shown by comparing Figure 6-14b and Figure 6-14c. Most of the smaller defects are removed, and only the large, significant ones remain. Performance data for classification with this strategy is recorded in Table 6-1 and Table 6-2 as data strategy ‘g’. As in data strategy ‘e’, all image windows comprising an individual defect are assumed identified if over half of them have been. Comparing data strategies ‘e’ and ‘g’ shows that removing these smaller defects results in fewer false negative predictions, but also fewer true positive predictions. This results in an improvement in sensitivity/recall for all classification strategies, though particularly for classification strategy 4 (mortar and brick are processed separately with contradicting regions blacked-out), which shows the best performance of all the classifiers. This suggests that this strategy is better suited to detecting the larger defects, potentially as parts of the smaller defects are blacked-out. However, lower precision and specificity for classification strategy 4 means that its overall performance scores are not the best of all the classifiers. All classifiers show a large fall in precision, and this is because the data is more imbalanced once smaller defective regions have been removed, meaning that false positive predictions have a much

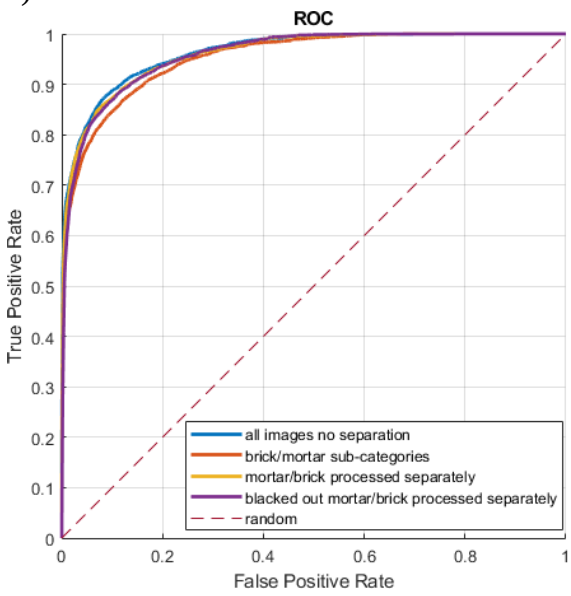
larger effect on the precision score as there are fewer true positive possibilities. Figure 6-13a shows the ROC curve when smaller defects are removed: performance improves for all classification strategies, with all strategies showing a similar performance.

Additional analysis has been performed as it is recognised that certain classes of defect are naturally smaller than others, and as such have a greater structural significance at a smaller size. Therefore, each defect class has been processed independently to remove defects with an area smaller than a proportion of the class mean, either ¼ or ¾. This means that for defects that tend to be smaller such as cracking, a smaller threshold is used so that smaller defects are retained. Where defects smaller than ¼ of the mean are removed, this equates to removing defects with area smaller than 11,689px for vegetation, 632px for spalling, 455px for mortar loss and 367px for cracking. Where defects smaller than ¾ the mean are removed, this equates to removing defects with area smaller than 35,068px for vegetation, 1897px for spalling, 1366px for mortar loss and 1102px for cracking. For reference, a header brick has an area of approximately 22,500px.

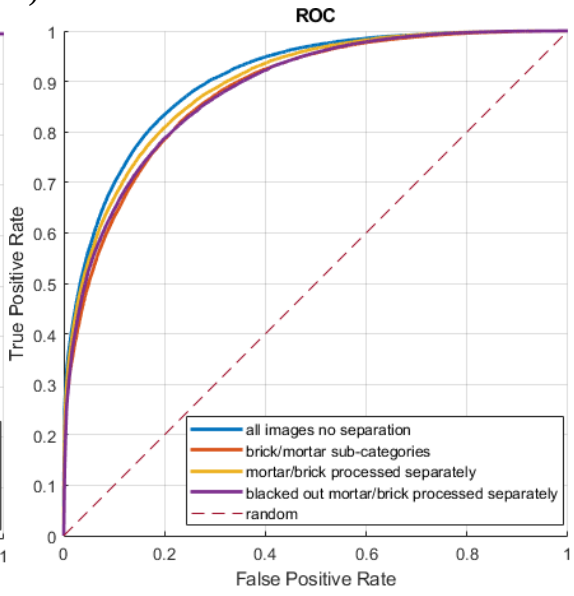
The effect of this processing strategy on the ground truth data for one wing-wall is shown in Figure 6-14d and Figure 6-14e, where more of the smaller, non-vegetation defects are retained than before in Figure 6-14c. Performance is shown in the ROC curves in Figure 6-13b and Figure 6-13c. For all classification strategies, lower performance is shown than Figure 6-13a, where more vegetation regions are included, but slightly better performance is shown than the dashed lines in Figure 6-7, where the only difference in the data strategy is that the smaller defects are not removed. This therefore suggests performance is better on larger defects.

For the section of wing-wall imaged in Figure 6-14a, and its ground truth in Figure 6-14e, the output of the different defect classification strategies are compared in Figure 6-15. Image windows predicted to be defective with a probability over 80% are highlighted in yellow. All the classification strategies show a similar output, which is to be expected considering all the performance data has suggested a similar performance. It is apparent that the major defects present in the image have been accurately identified, though less so for the smaller defects. Additionally, there are several seemingly random false positive predictions in all four strategies, though this appears more severe in Figure 6-15a, with classification strategy 1, where all image windows are processed together with no mortar/brick separation. This noise is particularly apparent in the lower ½ of the image, where looking at the raw image the wall has been painted with a patchy/peeling coating of black paint.

a)



b)



c)

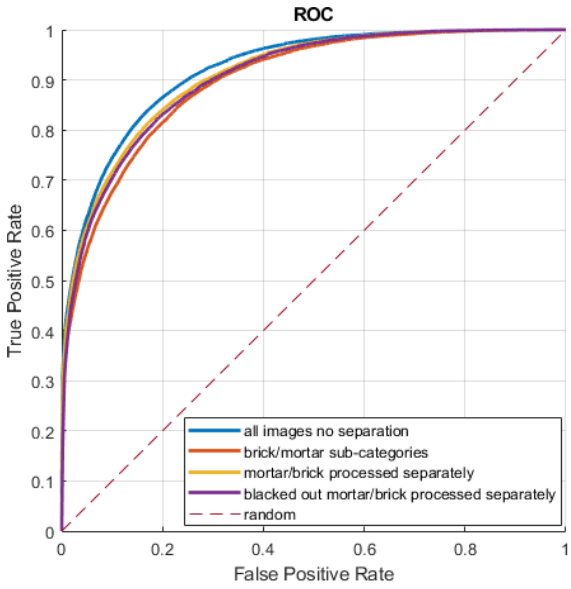
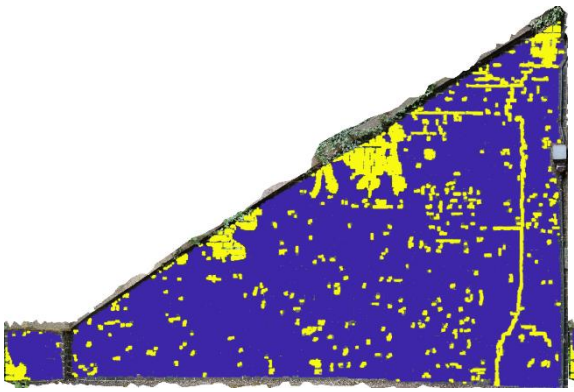


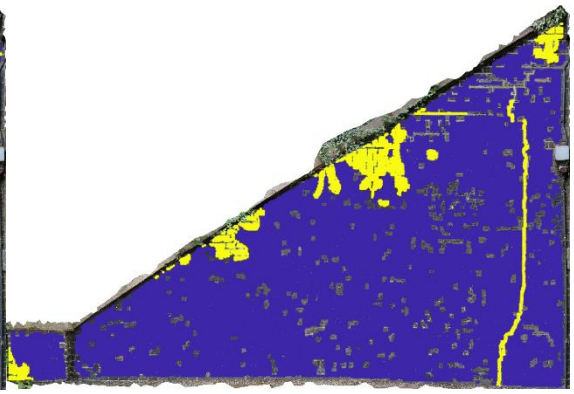
Figure 6-13: ROC plot for different masonry detection strategies assuming defect is detected if over half of its area is detected, showing: a) defects smaller than ⅓ header brick in size removed, b) defects smaller than ¼ the mean defect size for each class removed, c) defects smaller than ¾ the mean defect size for each class removed

a)

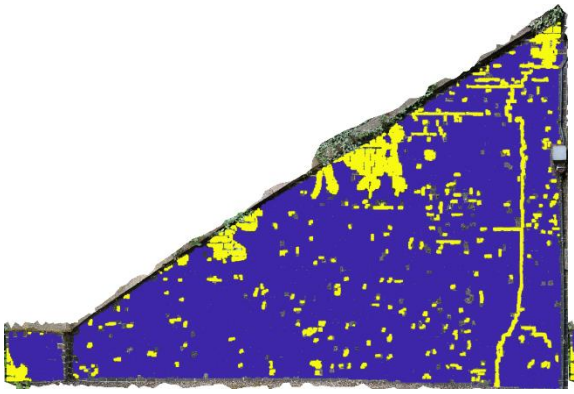
b)



c)



d)



e)

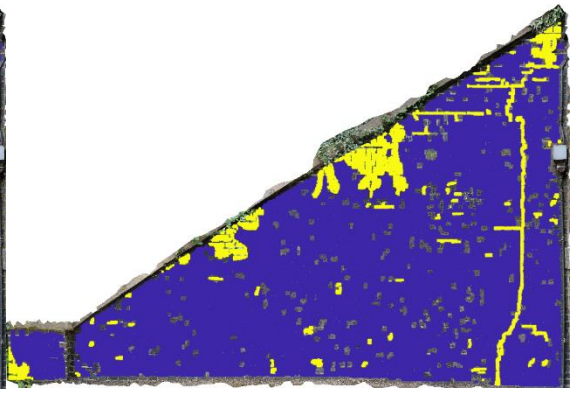


Figure 6-14: Ground truth data for a wing-wall section of bridge SBR4 showing: a) raw image, b) all defects, c) defects smaller than ⅓ header brick in size removed, d) defects smaller than ¼ the mean defect size for each class removed, e) defects smaller than ¾ the mean defect size for each class removed

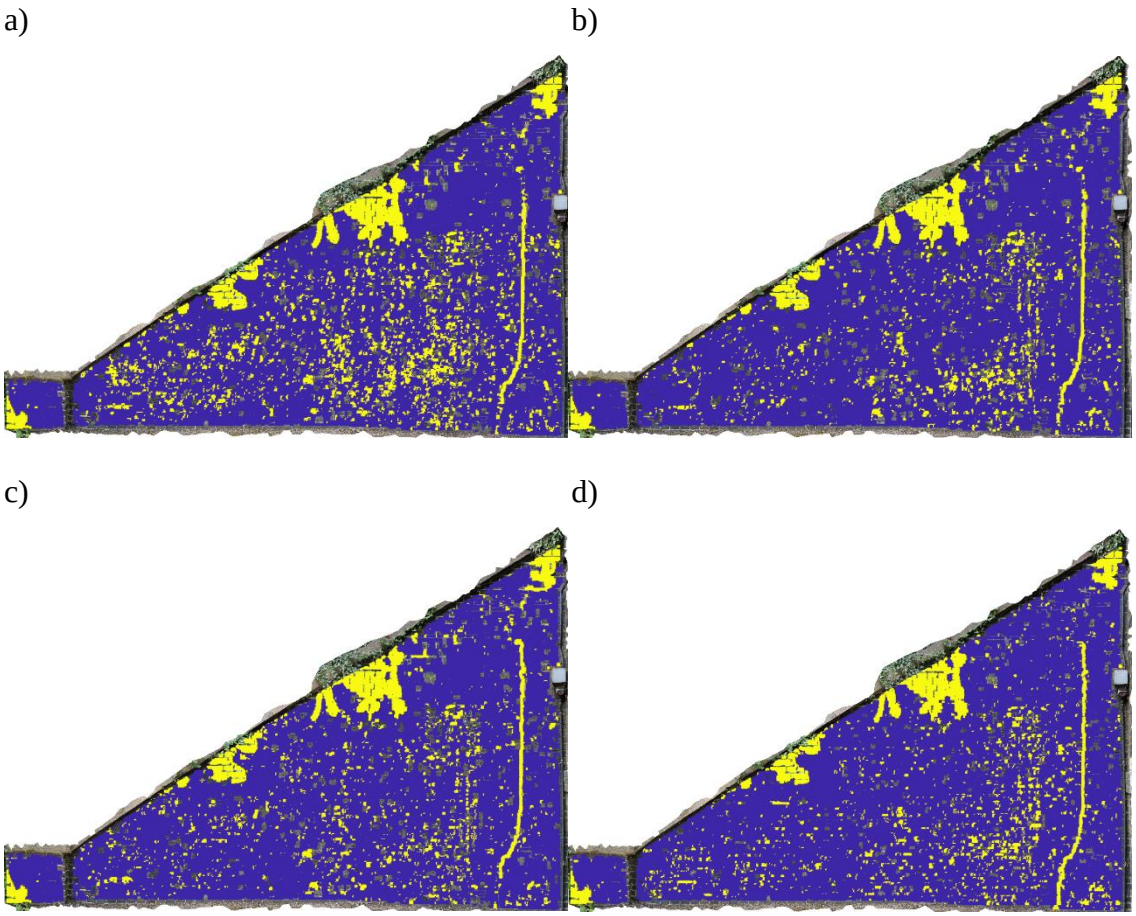


Figure 6-15: Image windows predicted with 80% certainty to contain a defect for a wing-wall section of bridge SBR4, with only defects greater than ¾ the mean size for each class retained showing the different defect detection strategies: a) all images with no separation, b) brick/mortar sub-categories, c) mortar/brick processed separately, d) blacked-out mortar/brick processed separately

6.1.2.2 Results from testing on dataset of viaduct images not used for training

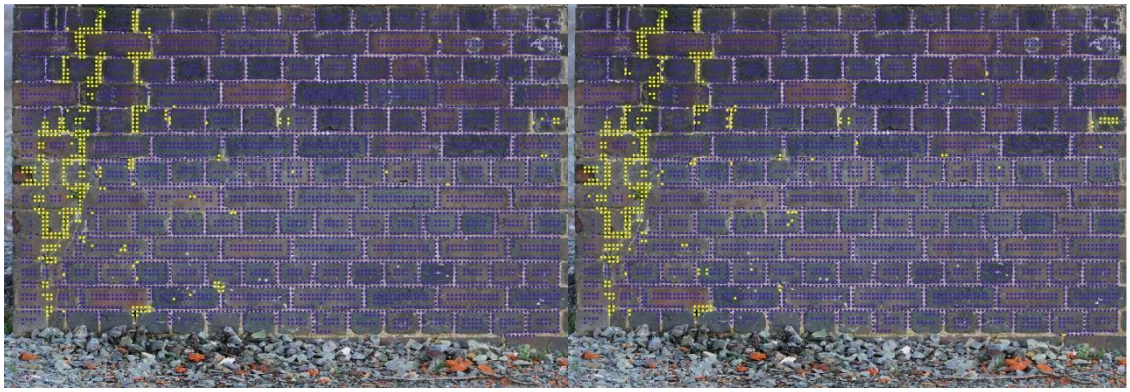
As described above in Chapter 6.1.1.1, a second test has been performed whereby the different classification methodologies were trained on a subset of 85 of the images in the training dataset, with the remaining 9 images used for testing, as it was recognised that the previous dataset used for testing contains bridge components not seen in the training dataset. In this case, the classifiers have therefore been tested on images from the same bridges, though the testing images were of separate parts of the bridges than the training images.

Figure 6-16 shows the output of the different classification strategies for one image tested. Image windows predicted to be defective with a probability over 80% are highlighted in

yellow. All four classification strategies give a similar output, all very comparable to the ground truth data in Figure 6-16b. In comparison to the output in Figure 6-15, there are fewer background false positive predictions, potentially due to the examined surface having less noise and distractors.

a) b)

c) d)



e) f)

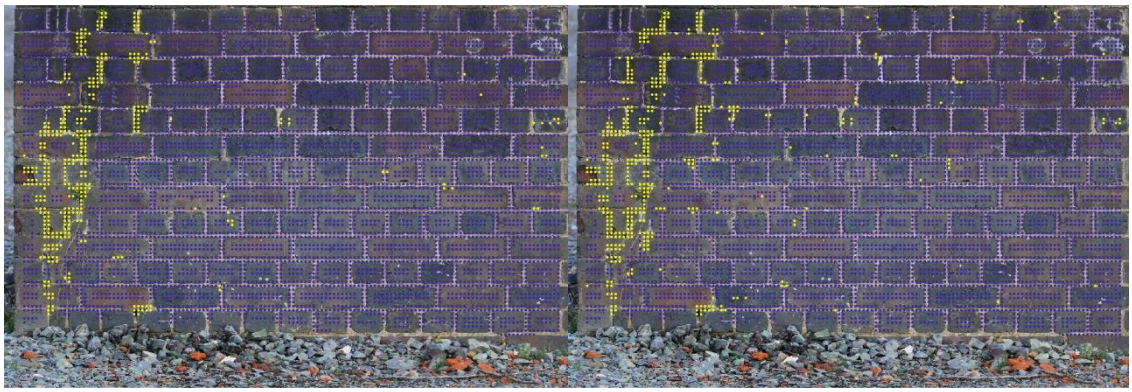


Figure 6-16: Image windows predicted with 80% certainty to contain a defect for an image from the viaduct test dataset showing the different defect detection strategies: a) raw image, b) ground truth, c) all images with no separation, d) brick/mortar sub-categories, e) mortar/brick processed separately, f) blacked-out mortar/brick processed separately

Performance results are presented in Table 6-3 and Table 6-4, in the same format as for the previous results, with both a confusion matrix and derived performance metrics. Looking at data strategy ‘a’, the raw output of the different classifiers, it is clear from the confusion matrix that all the classification strategies except classification strategy 4 perform very similarly. Recall that for classification strategy 4 the brick areas of image windows are blacked-out if the image window is mostly in a mortar region and vice versa. The cause of the lower performance for classification strategy 4 has been examined above, caused by the prevention of potential defects from being classified due to being blacked-out. The performance metrics in Table 6-4, and the ROC curve in Figure 6-17a, also show the same conclusions regarding the relative performance of the classification strategies.

Table 6-3: Confusion matrix comparing the performance of the different classification strategies for defect detection in masonry with testing dataset from viaduct images not used for training

				Classification strategies							
				1	2	3	4	1	2	3	4
				Predictions							
				Defective				Clean			
Data strategies	a	Actual	Defective	6930	6892	6950	6632	530	568	510	828
	b			774	786	788	766	69	57	55	77
	c			6156	6106	6162	5866	461	511	455	751
	a		Clean	3714	3043	4045	4712	53,230	53,901	52,899	52,232
	b			827	735	969	821	20,039	20,131	19,897	20,045
	c			2887	2308	3076	3891	33,191	33,770	33,002	32,187

Table 6-4: Performance metrics comparing the performance of the different classification strategies for defect detection in masonry with testing dataset from viaduct images not used for training

Performance metric	Data strategies	Classification strategies			
		1	2	3	4
Specificity	a	0.935	0.947	0.929	0.917
	b	0.960	0.965	0.954	0.961
	c	0.920	0.936	0.915	0.892
Sensitivity/recall	a	0.929	0.924	0.932	0.889
	b	0.918	0.932	0.935	0.909
	c	0.930	0.923	0.931	0.887
Precision	a	0.651	0.694	0.632	0.587
	b	0.483	0.517	0.449	0.483
	c	0.681	0.726	0.667	0.601
F1 score	a	0.766	0.792	0.753	0.705
	b	0.633	0.665	0.606	0.631
	c	0.786	0.813	0.777	0.717
Balanced accuracy	a	0.932	0.935	0.930	0.903
	b	0.939	0.949	0.944	0.935
	c	0.925	0.929	0.923	0.889
Brier score	a	0.052	0.046	0.056	0.069
	b	0.030	0.029	0.036	0.032
	c	0.063	0.055	0.067	0.088

Classification strategies: (1) no mortar & brick separation, (2) mortar & brick labelled as separate categories, (3) mortar & brick regions processed separately, (4) mortar & brick regions processed separately and blacked-out.

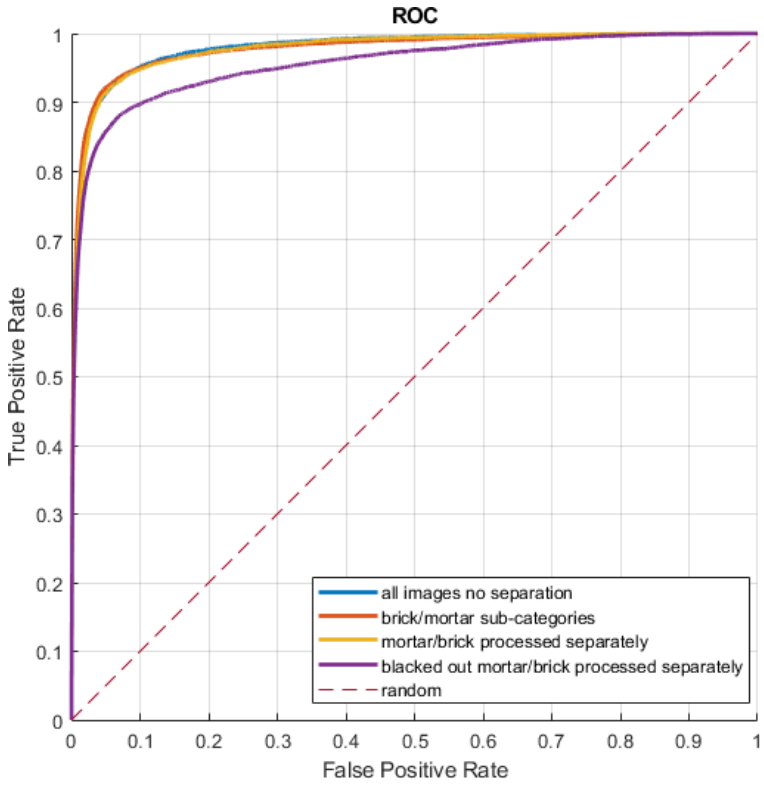
Data strategies: (a) Raw data, (b) only brick regions, (c) only mortar regions

Comparing the performance metrics in Table 6-4 with those in Table 6-2, and the corresponding ROC curves in Figure 6-17a and Figure 6-7, the performance of the classifiers on this dataset is far superior to the performance on the dataset of the 3 independent bridges tested previously. This could be due to the cleaner masonry present in this dataset. The previous dataset included the arch barrels of the bridges, which have more noisy surface textures than other parts of a bridge both due to the way they are constructed, and due to the increased exposure to elements such as wetness leading to more surface deposits. As the previous dataset included complete bridges, it also included masonry surfaces where the presence of defects is less clear cut, whereas the viaduct images dataset used here includes selected images based on their depiction of defects. Additionally, comparing bridges SBR4, SBR8 and SBR12 to the viaduct bridges in Chapter 4.2, the distractors present are greater and more frequent. Finally, the higher performance could be due to the masonry colours and construction detail being similar to

that seen in training, given that the test data and the training data are from the same bridges. This would suggest that including even more bridges with an even larger variation of masonry texture in the training dataset could further improve generalisation.

Like before, the output of the classifiers has been filtered such that the performance of the brick and mortar regions can be compared individually. These results are shown in data strategies ‘b’ and ‘c’ respectively in Table 6-3 and Table 6-4, and the ROC curves in Figure 6-17b. The ROC curves show the performance of all classifiers to be better in the brick regions than the mortar regions, though by a smaller margin than before. This smaller margin is a result of better performance in both regions, suggesting that with cleaner masonry conditions, there is less of a marked difference between performance in mortar and brick regions.

a)



b)

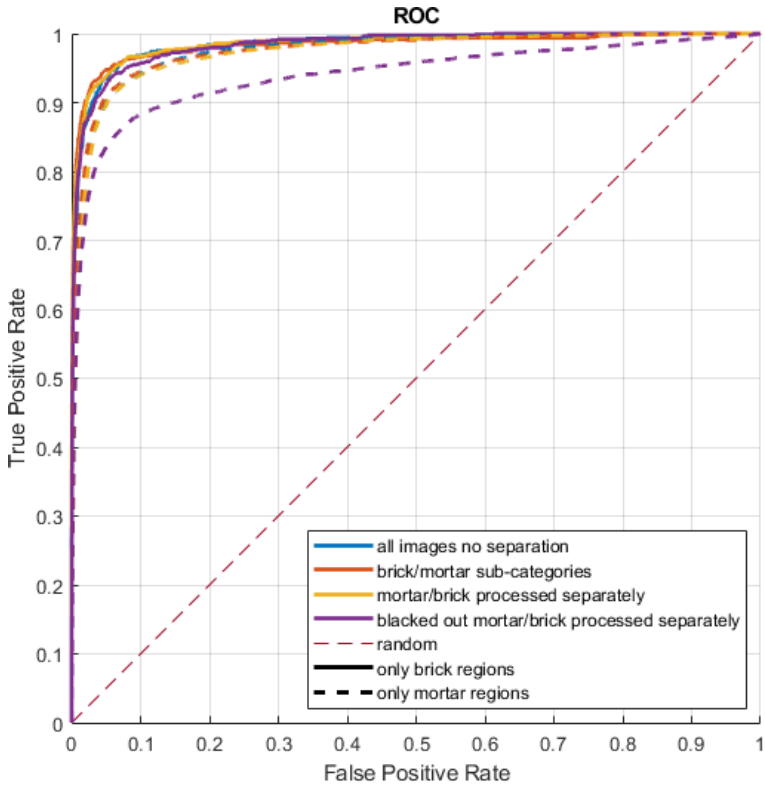


Figure 6-17: ROC plot comparing different masonry detection strategies with testing dataset from viaduct images not used for training: a) raw data, b) mortar and brick regions separately

6.1.2.3 Class Activation Mapping

Class Activation mapping (CAM) is a technique that can be used to localise the regions of an image that most strongly cause a given class to be predicted. It therefore gives a partially semantic classification output despite the original classifier being trained on non-semantic, image window labelled data. This means the location of features within the image is inferred rather than learned by the classifier. CAM can also be used to help understand what information a classifier is using to differentiate between classes, making the classifier less of a 'black box'. It is for this reason that CAM is used in this work, to establish whether the classifier is learning features of mortar joints as defects.

CAM takes the output from the last convolutional layer of the network, the layer before the localisation data is lost in the fully connected layers used for classification. A heatmap of class activation is then derived from the gradients of the investigated class into this layer and the output feature map of the layer. In this way, the importance of each channel in the output feature map to the defect class being investigated is determined (Selvaraju et al., 2016).

Figure 6-18 presents different CAM outputs for the defective class on images in the viaducts testing dataset analysed in Chapter 6.1.2.2. In this dataset, it was observed that the performance of classification in mortar joint regions was worse than those in brick regions, though to a smaller extent than in the three independent bridges dataset analysed in Chapter 6.1.2.1. Figure 6-18 shows such mortar regions of the tested images. In Figure 6-18a, where a correct, non-defective prediction was given, there is little activation giving weight to a defective classification. However, in the 3rd and 4th images of Figure 6-18a, a minor activation is focused on the mortar joint, specifically the dark interface between the mortar and the brick. The CAM images depicting image patches which were incorrectly predicted as defective in Figure 6-18c all show that the cause of the incorrect defective classification is regions on the mortar joint. Specifically, the cause of incorrect classification is either dark shadowy regions (1st, 2nd, and 5th images), or much thinner more crack-like features on the mortar/brick interface (3rd and 4th images). In Figure 6-18b, patches that have been correctly classified as defective are shown. In this case, some of the defects are purely in the mortar regions while others span across from the mortar regions into the brick regions. In all cases, the defect location has been correctly identified. This is despite the mortar/brick interface also showing some shadowy regions like those causing incorrect false positive classifications in Figure 6-18c. This therefore

suggests that the strength of the activation and hence classification for shadowy regions on the mortar interface is less than that caused by actual defects on the surface.

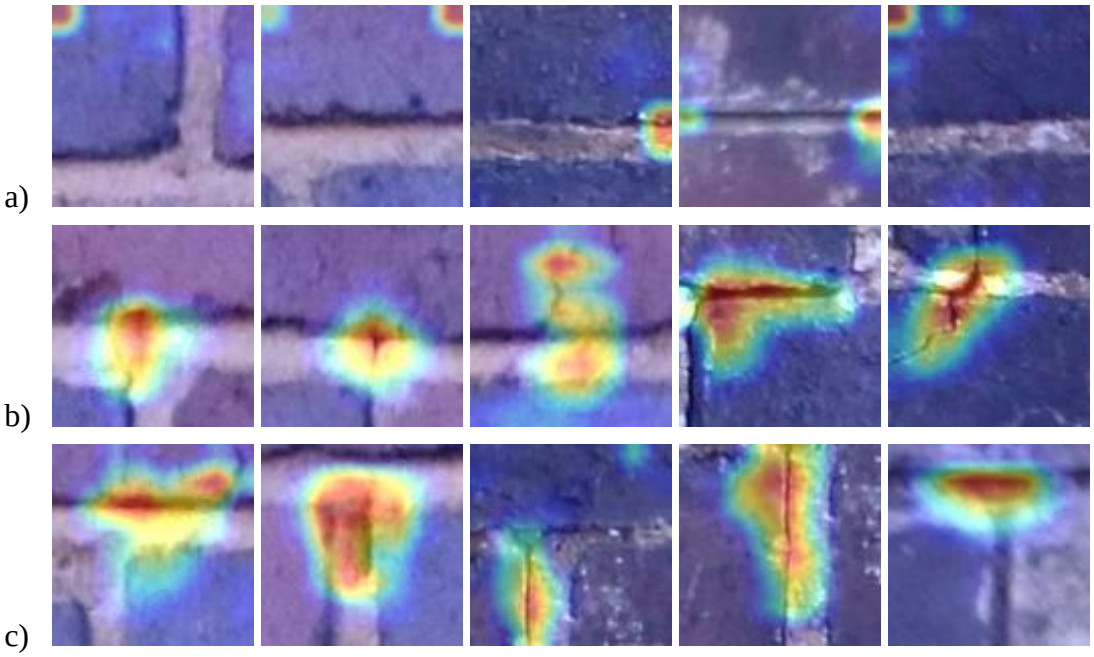


Figure 6-18: CAM output for the defective class on different image patches depicting mortar joints where image patches are: a) correctly predicted clean, b) correctly predicted defective, c) incorrectly predicted defective

6.2 Chapter Summary

This chapter has sought to answer the following key research question of this thesis: how do mortar joints on inherently noisy masonry surfaces affect the ability of a state of the art deep learning classifier to classify defects.

Initially, an edge detection based defect detection algorithm was tested. It was shown that with edge detection alone, it would be impossible to detect defects in masonry, as the mortar interface causes a much greater response to the filter. Therefore, the developed algorithm focused on filtering out and masking the mortar regions and then detecting defects in only the brick regions. The potential defects detected in the unmasked brick regions were then filtered through grouping and then thresholding the smaller detections. This gave an output which highlighted all inconsistencies in the masonry, including defects, writing, and irregular mortar joints. This algorithm was tested on a relatively simple masonry wall with comparatively little noise.

A deep learning algorithm was therefore investigated to deal with more noisy, less uniform masonry surfaces. A GoogleNet Inception v3 classifier was trained and tested

for classifying image windows generated from the bridge and viaduct datasets created in Chapter 4. Deep learning is much more powerful than edge detection, and therefore it was investigated if the same conclusion would hold, i.e., that the mortar joints make defect detection more difficult. To test this previous conclusion, four different defect detection strategies were used: 1) all image windows classified together with no mortar and brick separation, 2) mortar and brick regions labelled as separate categories within one classification, 3) mortar and brick regions processed and classified separately, and 4) mortar and brick regions processed and classified separately with the contrasting regions blacked-out. It was found that all four classification strategies had a similar performance, with the performance spread between the different bridges and images tested being far greater than that between the different classifiers.

Different data strategies were then investigated, including assuming a defect is identified if half of its area has been identified, removing vegetation regions as they occlude mortar joints, and removing smaller defects to investigate the detection of larger and more structurally significant defects. It was found that these data strategies altered the performance of all classification strategies, though the relative performance between the different strategies remained similar.

These findings therefore suggest that a powerful deep learning classifier can learn the features of the mortar and brick interface in masonry so as not to confuse them with defects. However, an additional data strategy was also performed, separating the classification results of those image windows containing only brick from those containing mortar, and it was found that the classification performance of mortar regions was worse than brick. This therefore suggests that the mortar regions are the more challenging locations for the classifier to detect defects.

The different classification strategies have also been tested on a second dataset, this one more like the training data, with less surface noise on the masonry than the previous dataset. It was found that the performance of all classifiers was better on this dataset, though they all again showed a similar performance. Testing on the mortar and brick regions separately showed a better performance on the brick regions, though by a much smaller margin than on the noisier dataset. This suggests that on less noisy masonry images, more like the training data, classifying defects in mortar regions compared to brick regions is comparatively less of a challenge than with the noisier dataset.

Class activation mapping was used to determine the areas of each image window that the classifier most strongly associates with the defect class. Image windows which were correctly classified as non-defective generally showed little response, though for those that did show a more pronounced response, this was around dark regions on the mortar/brick interface. In image windows incorrectly classified as defective, the response heatmap suggested that classification was often caused by the dark shadows and narrow crack like features at the mortar/brick interface. However, in image windows that were correctly classified as defective, the heatmap was able to highlight the defect, despite shadows appearing on the interface, suggesting the actual defect caused a far greater response in the classifier.

These findings confirm that when capturing masonry surfaces for classification, it is important to ensure that the masonry surface is well lit with neutral, direct lighting to avoid shadows on the mortar/brick interface where the mortar is recessed, as these shadows have the potential to look like defects.

7 EXPERIMENTAL EVALUATION OF AUTOMATED DEFECT DETECTION METHODOLOGY IN COMPARISON TO MANUAL INSPECTION

7.1 Manual inspection dataset generation

A benchmark for the performance of an automated classifier is to examine its performance against that of the current defect identification methodology for masonry arch bridges, namely manual inspection. The existing manual inspection procedures as well as their known subjectivity are discussed in detail in the literature review, in Chapters 2.1.2 and 2.1.3. It is already assumed that an automated defect classification methodology alongside an automated data capture methodology would result in safety and logistical performance benefits compared to manual inspection. This is due to avoiding the need for an inspection engineer to closely examine structures from height, often alongside and potentially causing the temporary closure of major infrastructure corridors. Therefore, this work solely focuses on the detectability of defects using manual inspection compared to an automated procedure.

A dataset of the defects identified during manual inspection has been created based on the dataset created in Chapter 4 for the following bridges: Spring Road Bridge (SBR 4),

Norton Road Bridge (SBR 8), and Smythes Farm Bridge (SBR 12). These are the bridges which were used to test the automated classifiers in Chapter 6.1.2.1. The manual inspection data for these bridges has been annotated in a similar fashion to the annotation of the ground truth data in Chapter 4.6. The data of defects identified during manual inspection has been attained from the most recent inspection report for each bridge, respectively. These reports are all from detailed inspections, the highest standard of inspection that is routinely performed on bridges, performed by three different bridge inspectors. The inspections were undertaken in December 2016 for bridge SBR4, January 2018 for bridge SBR8 and February 2016 for SBR12. As the image data from which the ground truth and automated detection data is derived was captured in June 2019, there is a gap of up to 3.5 years between the inspection and ground truth datasets. However, this is thought not to be problematic since, aside from vegetation, the other defect categories being sought tend to be long lived, so are unlikely to have changed in the interim. Indeed, additional graffiti has been applied to some of the bridge surfaces between the two data capture dates, resulting in harder defect detection conditions in the automated detection data than was present during manual inspection.

While recording the defects identified through manual inspection it was observed that for some bridges, despite the presence of a defect being mentioned in the report, its precise location on the structure was often omitted, as were images qualifying the defect. This is detailed for each bridge in the list below. Additionally, defects often were not individually identified. For example, terminology such as ‘numerous vertical fractures open to 6mm’ was used for describing a particular element of the bridge. In these cases, the benefit of doubt has been given, and all defects in the ground truth data matching the description have been marked as detected by manual inspection.

By bridge, the breakdown of the documentation quality in the inspection reports is:

- Spring Road Bridge (SBR 4): Only 55% of the defects identified in the report have been photographed. Additionally, there is no plan drawing of the bridge, or its element detailing, where each of the identified defects are located.
- Norton Road Bridge (SBR 8): Much more in depth detailing of defects than either of the other two bridges. However, only 38% of the defects identified in the report have been closely photographed, though a few others are visible, though not discernible in overview photographs of the structure.

- Smythes Farm Bridge (SBR 12): Well documented report with all the identified defects imaged, and defect locations annotated on a plan of the bridge.

7.2 Detection accuracy evaluation

It was observed in Chapter 6.1.2, that all the different defect detection strategies tested yielded a similar performance. Therefore, no prior separation of mortar joints is performed here. The classifier trained for classification strategy 1 detailed in Figure 6-5 in Chapter 6.1.1.1, with no mortar and brick separation, is used and tested here.

As detailed in Chapter 6.1 this is a classifier based on the GoogleNet Inception v3 architecture that has been trained to classify image windows of masonry into defective and non-defective categories. This has been trained with transfer learning using data from the dataset described in Chapter 4.6 comprising images from mostly viaducts.

The test dataset is the same as the dataset used in Chapter 6.1.2.1, though in this case all image windows are included (before image windows which depicted half mortar, and half brick were excluded). The resulting testing dataset now consists of 2,708,015 image windows. The class distribution of these image windows is the same as that of the bridge surfaces, as this is testing the classification across the whole bridge, as would be done in practice. There therefore is a large imbalance between the defective and clean classes, with 438,106 defective and 2,269,909 clean image windows.

7.2.1 Results

As in Chapter 6.1.2, the performance of the classifiers has been presented in a confusion matrix in Table 7-1, with performance metrics in Table 7-2, and using ROC curves in Figure 7-1 to Figure 7-4. Explanation of the calculation and meaning of the different performance metrics used is given in Chapter 6.1.1.2. Here, it is worth reiterating that all the metrics other than the Brier score and the ROC curve do not account for the confidence of a particular classification. They rather score a binary classification based on a prediction being above or below a threshold of 0.5.

Firstly, data strategy ‘a’ which represents the raw data is examined. In the confusion matrix, the automated classification strategy yields far more true positive window classifications when compared to manual classification. This means that many more defect instances have been classified, and fewer have been missed. Conversely, many more false-positive window classifications have been made by the automated classification strategy compared to manual. This is inherent to the methodology, as no

false-positive window classifications were made by manual classification, as the manual inspection mask was defined relative to the ground truth mask. Regardless, manual inspection classification is likely to contain few false positives as an inspection engineer is only likely to record defects with high certainty.

Table 7-1: Confusion matrix comparing the performance of the developed automated classifier with that of manual inspection on 3 masonry arch bridges

				Classification strategies			
				Automated classification	Manual inspection	Automated classification	Manual inspection
				Predictions			
				Defective		Clean	
Data strategies	a	Actual	Defective	280,904	134,898	157,202	303,208
	b			162,567	27,618	147,228	282,177
	c			323,533	143,020	114,573	295,086
	d			197,001	32,961	112,794	276,834
	a		Clean	278,816	0	1,991,093	2,269,909
	b			278,816	0	1,991,093	2,269,909
	c			278,816	0	1,991,093	2,269,909
	d			278,816	0	1,991,093	2,269,909

Data strategies: (a) Raw data, (b) excluding vegetation regions, (c) assume defect is identified if more than 50% is identified, (d) excluding vegetation regions and assume defect is identified if more than 50% is identified.

This results in performance metrics showing perfect precision and specificity for manual inspection, as would be expected. Precision is much lower for automated classification due to many false positive classifications of defects. Specificity is not affected as much as precision due to the class imbalance – there are many more negative (non-defective) cases, and the majority of these were correctly classified. Sensitivity and recall on the other hand show much better performance for the automated classification methodology relative to manual inspection, due to the greater proportion of defective instances detected. This results in better balanced accuracy and F1 scores for automated classification. However, the Brier score, also a measure of overall performance which additionally accounts for the relative certainty of predictions, shows similar performance between manual and automated methodologies, with the manual performance slightly

superior. Given most image windows in the tested dataset show no defects, and the image windows in manual inspection are predicted with 100% confidence (either 0% or 100% likelihood of being defective), the brier score therefore biased towards the manual classification output.

Table 7-2: Performance metrics comparing the performance of the developed automated classifier with that of manual inspection on 3 masonry arch bridges

Performance metric	Data strategies	Classification strategies	
		Automated classification	Manual inspection
Specificity	**a**	0.877	1.000
	b	0.877	1.000
	c	0.877	1.000
	d	0.877	1.000
Sensitivity/recall	**a**	0.641	0.308
	b	0.525	0.089
	c	0.739	0.327
	d	0.636	0.106
Precision	**a**	0.502	1.000
	b	0.368	1.000
	c	0.537	1.000
	d	0.414	1.000
F1 score	**a**	0.563	0.471
	b	0.433	0.164
	c	0.622	0.492
	d	0.502	0.192
Balanced accuracy	**a**	0.759	0.654
	b	0.701	0.545
	c	0.808	0.663
	d	0.757	0.553
Brier score	**a**	0.122	0.112
	b	0.125	0.109
	c	0.108	0.109
	d	0.113	0.107

Data strategies: (a) Raw data, (b) excluding vegetation regions, (c) assume defect is identified if more than 50% is identified, (d) excluding vegetation regions and assume defect is identified if more than 50% is identified.

Therefore, it can be concluded that both the manual and automated strategies give a similar level of performance, with the automated strategy detecting a greater proportion of the defects present, but the manual strategy leading to far fewer false positive detections. However, in a safety critical application such as detecting defects in masonry arch bridges, arguably ensuring all defects are identified is more important than the

number of false positive detections. Nevertheless, too many false positives could cause extensive manual interpretation of results and could render an automated method too time consuming for practical use.

The ROC curve in Figure 7-1 mirrors these same conclusions, with the automated defect classification methodology giving both a higher true positive rate and a higher false positive rate.

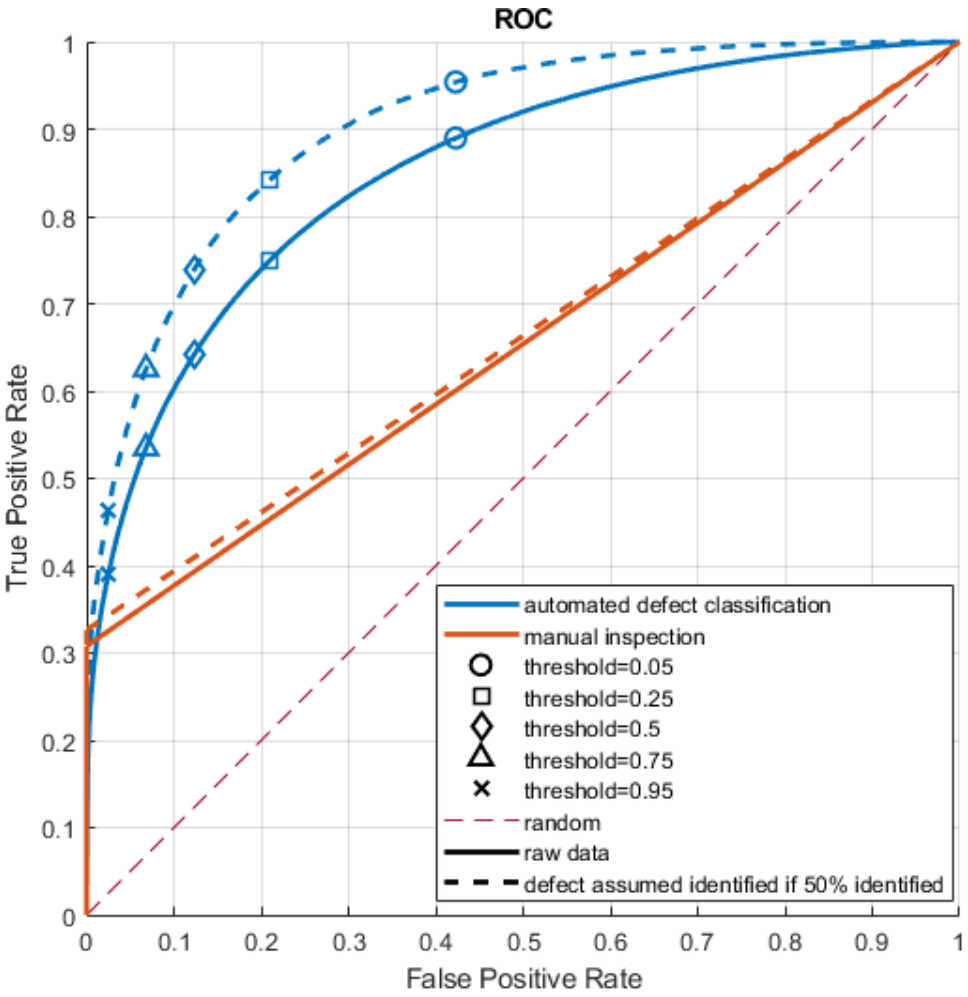


Figure 7-1: ROC plot comparing automated vs manual defect detection in masonry arch bridges

As the manual inspection classifications have been determined exactly, either with 0% or 100% probability of a defect being present, the manual inspection curve is only plotted from three datapoints. Two of these are end points at (0,0) with a theoretical threshold of 100% where no image windows are classified as defective and at (1,1) with a theoretical threshold of 0% where all image windows are classified as defective. The final datapoint is that corresponding to the actual performance of the classifier at all other thresholds.

The automated classifier made predictions with a range of different confidences resulting in a continuous curve between the two end points. Thresholds against these different prediction confidences have also been plotted in Figure 7-1. At a threshold of 0.95 (i.e., only classifying those image windows as defective which have been predicted as defective with more than 95% confidence), the performance of the automated classifier is much more comparable to manual inspection, in terms of both the false positive and true positive rates.

Individual vegetation defective regions usually tend to be comparatively large, so have an outsized impact on performance statistics which are based on defective area detected. Additionally, these regions are those most likely to have significantly changed in the period between the dates of manual inspection and ground truth data capture. Finally, in manual inspection reports, vegetation regions were the defect type described with the greatest amount of ambiguity, meaning benefit of doubt was most often given here and vegetation regions marked as detected. Statistics based on the vegetation data are therefore potentially the least dependable.

Figure 7-2 and data strategy ‘b’ in Table 7-1 and Table 7-2 shows the impact of removing vegetation regions from the dataset, therefore detailing the classification performance on only the other defect classes. Across the different performance metrics, this results in a decrease in performance for both automated classification and manual inspection. This should be expected as vegetation regions are potentially easier to detect in comparison to other defect classes due to the often-clear contrast with the masonry background. With vegetation regions excluded, manual inspection shows better precision and specificity, but worse sensitivity and recall than the automated classifier. Overall accuracy measures of balanced accuracy and particularly F1 score, show a greater decrease in performance with the manual inspection data than is seen with the automated classifier. This results in both scores showing significantly better performance with automated classification than with manual inspection. The Brier score however still shows a slightly better score for the manual classification. As described previously, this is a result of the severe class imbalance in the testing dataset resulting in the manual inspection being fully correct with predictions for the vast majority of the dataset (all non-defective image windows). Comparing the ROC curves in Figure 7-2 and Figure 7-1 also quantifies the relatively larger fall in performance for manual inspection when not including the vegetation regions. True positive rate predictions (the proportion of defect instances found) approximately decrease from 0.3 to 0.09. Therefore, at a threshold of 0.95, the

performance of the automated classifier almost matches that of the manual classifier in terms of its false positive rate, though far exceeds it for the true positive rate.

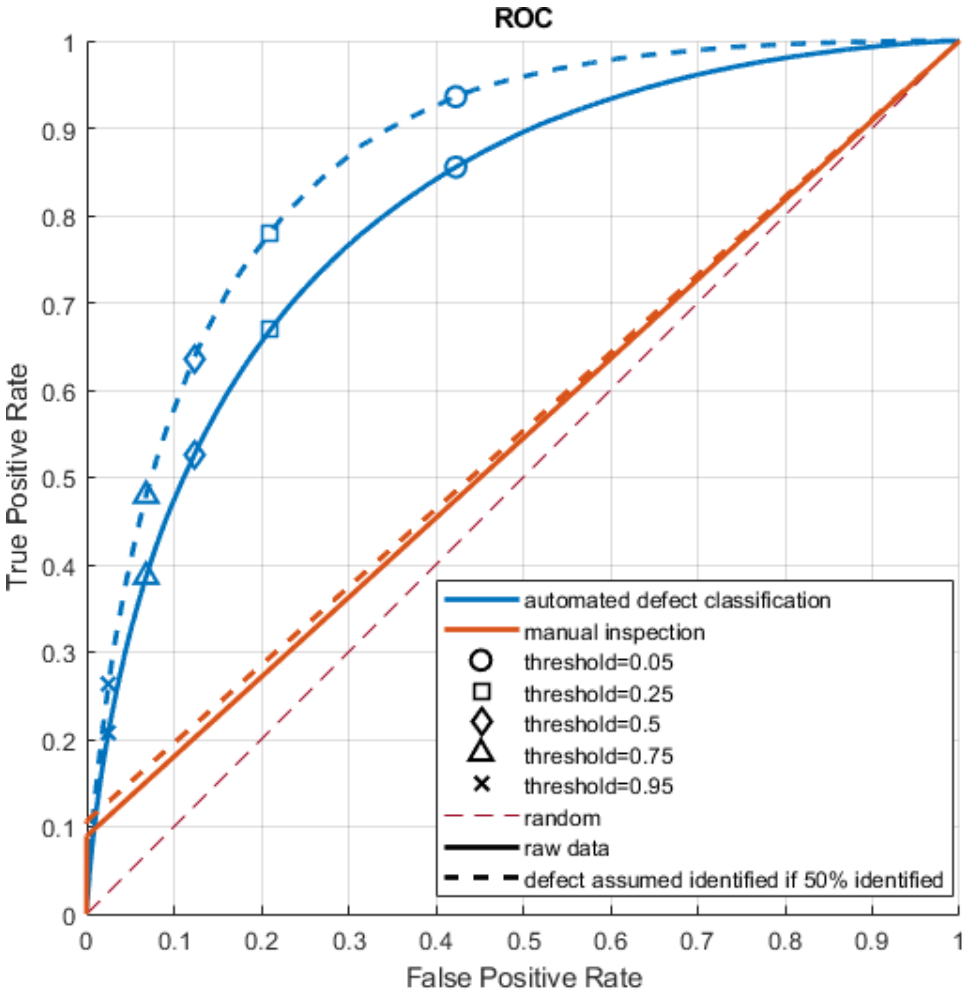


Figure 7-2: ROC plot with vegetation regions removed comparing defect detection performance

In both Figure 7-1 and Figure 7-2, and in data strategies ‘c’ and ‘d’ in Table 7-1 and Table 7-2, performance has been assessed based on the assumption that a unique defect has been identified if over half of the image windows depicting it have been classified as defective. This assumes that a bridge inspector would find almost equal utility in a tool that presented most of the area of a defect compared to one which presented its entire area. The methodology for calculating this performance statistic follows the same logic as that described in Chapter 6.1.2. As expected, this results in a performance improvement, though more significantly so for the automated classifier than for manual inspection data. The smaller performance improvement seen here for manual inspection is partially a result of the benefit of doubt being given when determining which defective areas have been detected through manual inspection. Therefore, the area of defects marked as

detected in inspection is likely to closely align with the ground truth data in cases where defective areas in inspection reports are loosely defined. Additionally, during an inspection, an inspector is likely to follow the path of a defect to its ends to identify it fully, meaning that once some of its area is established, it is likely that the whole defect will be established. This is not the case for the current implementation of the automated defect classification methodology whereby each image window is classed as defective or not based purely on its own merits.

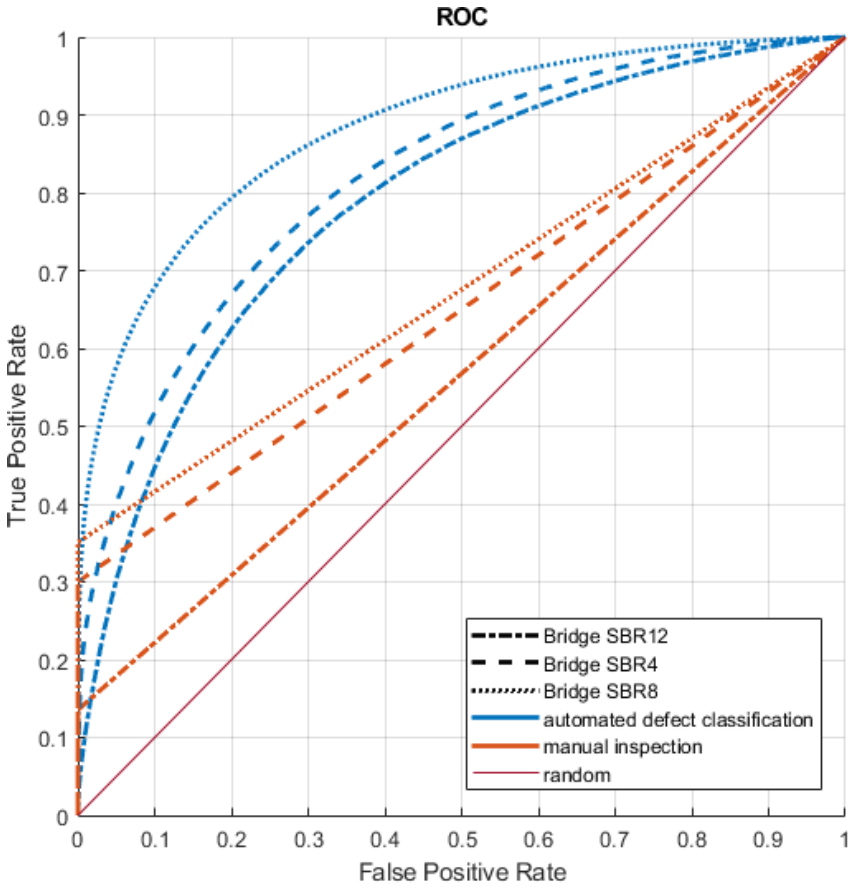


Figure 7-3: ROC plot comparing defect detection performance across the individual bridges tested

Current plots and statistics examined all show the average performance of the methodologies across all the data from the three bridges tested. This performance data is now broken down into the individual bridges in Figure 7-3, and the individual images tested, each representing an element of a bridge in Figure 7-4. Figure 7-3 shows that the relative performance of both the automated classification and manual inspection was better on the same bridges, with SBR8 showing the best performance and SBR12 showing the worst performance in both cases. This suggests that similar features on bridges are problematic for both automated classification and manual inspection. These potential

features were discussed in Chapter 6.1.2.1. Figure 7-4 shows that there is significant spread in performance between the individual images tested. This performance spread is greater for manual inspection than for automated classification. This suggests that some of the cause of the performance spread for manual inspection is due to human factors, such as height, discussed in Chapter 2.1.3, rather than purely masonry and image conditions on the bridge elements.

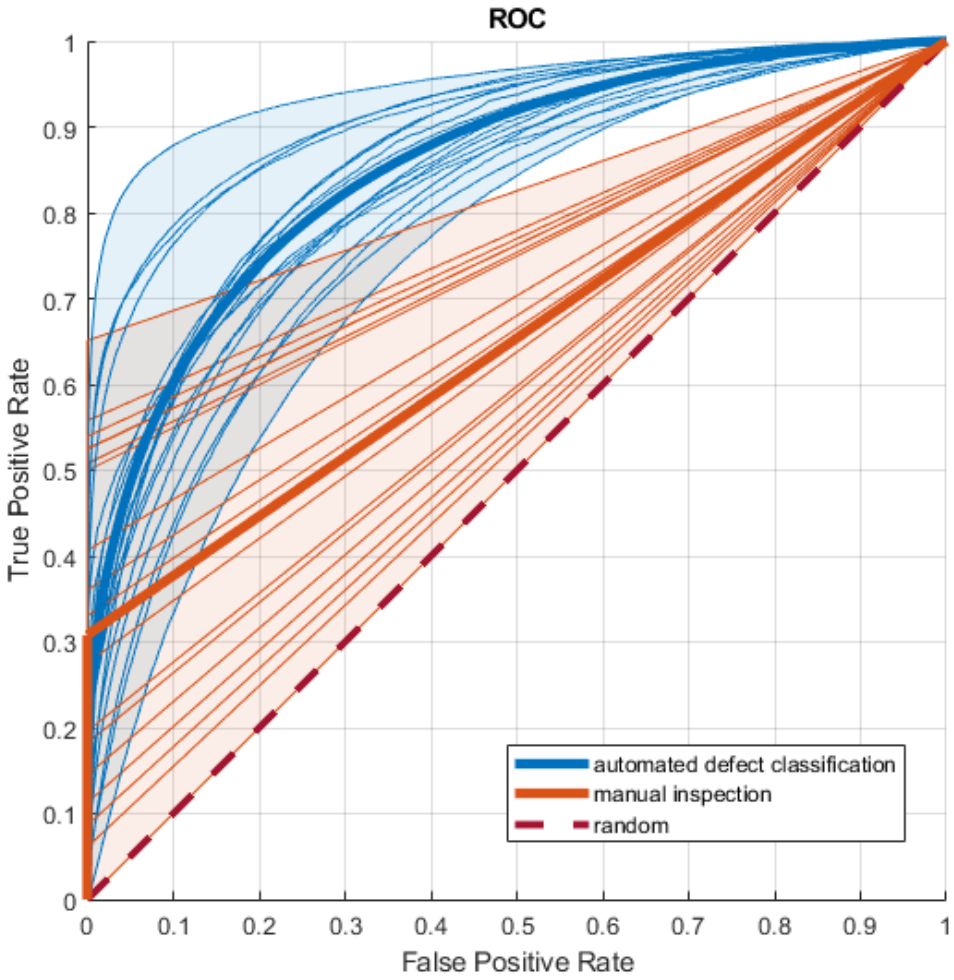


Figure 7-4: ROC plot comparing defect detection performance across the individual images tested

In Figure 7-5 to Figure 7-8, outputs for different bridge elements have been presented. Firstly, Figure 7-5 shows the performance of detection across the arch barrel and piers of bridge SBR12. This was determined to be the worst performing bridge for both manual inspection and automated classification in Figure 7-3. In this image, classification for both automated classification and manual inspection is poor, with most defects missed. Additionally, the automated classifier has mistaken the black border of the graffiti writing for a defect. The bridge surface here shows lots of surface deposits. The bridge piers are

caked in slurry deposits from passing farm traffic, and the arch barrel deposits are caused by efflorescence and other water borne deposits. These deposits somewhat mask defects present on the bridge surface, as well as creating a surface texture such that defects are less distinguishable from the background. Additionally, for the automated classifier, the training dataset did not include arch barrel masonry – only including masonry from the flat surfaces of the training bridges. The training dataset therefore did not include examples of efflorescence which tend to exclusively appear on bridge arch barrels, nor slurry deposits.

Figure 7-6 shows one of the front spandrel and parapet walls of bridge SBR4. The performance of the automated classifier is far better than in Figure 7-5, though manual inspection still misses many defects. In the centre of the image, there is bright sunlight, with shadowing towards the sides. The classifier is not confusing the shadow interface on the left for a defect, but some parts of the shadow interfaces caused by vegetation on the right are confused. Far fewer false positive detections are detected in the shadow region on the left of the image, on the spandrel wall and pier surface, than on the remainder of the masonry surface exposed to bright sunlight. False positive detections due to bright sunlight are particularly prevalent on the parapet walls in the sun. The mortar channel is deeper here than on the rest of the bridge resulting in shadowing which is mistaken for a defect.

Figure 7-7 shows a wing-wall of bridge SBR8. In this image, the performance of the automated classifier agrees very strongly with the ground truth data, and the performance of the manual inspection output also surpasses that seen in the previous two examples. The image in this example is neutrally lit, showing no shadows, especially at the mortar joints. Additionally, many of the defects present here are larger in scale than those in the previous examples, making classification easier.

Finally, Figure 7-8 shows the arch barrel and piers of the skew arch bridge SBR8. The manual inspection has picked up some of the major defects present, but a few are omitted. The performance of the automated classifier is better on the bridge piers (at the top and bottom of the image), then on the arch barrel, showing better agreement with the ground truth data. The arch barrel on SBR8 is heavily covered in soot and other deposits, making defect detection exceedingly difficult, even for the manual identification of defects to create the ground truth dataset.

The examples presented all show the automated classifier achieving better or similar performance to manual inspection, agreeing with the performance statistics examined above. In all the examples, the automated classifier predicted a greater proportion of the defects present, though also falsely predicted a greater proportion of the non-defective surface as defective – conclusions that were observed in the overall performance statistics above. In the presented examples, it was observed that the automated classifier performed worse in conditions that it was not given much exposure to in the training dataset, such as heavy sunlight, and surface deposits. In Chapter 6.1.2.2, the performance of the automated classifier was tested on a dataset more like the one used in training, with images coming from the same bridges, though of separate parts to that used in training. It was shown that the performance of the classifier was far superior on that dataset when compared to performance on the dataset tested in this chapter, suggesting that with additional training data, the automated classifier has the potential to far outperform the manual inspection for defect classification accuracy.

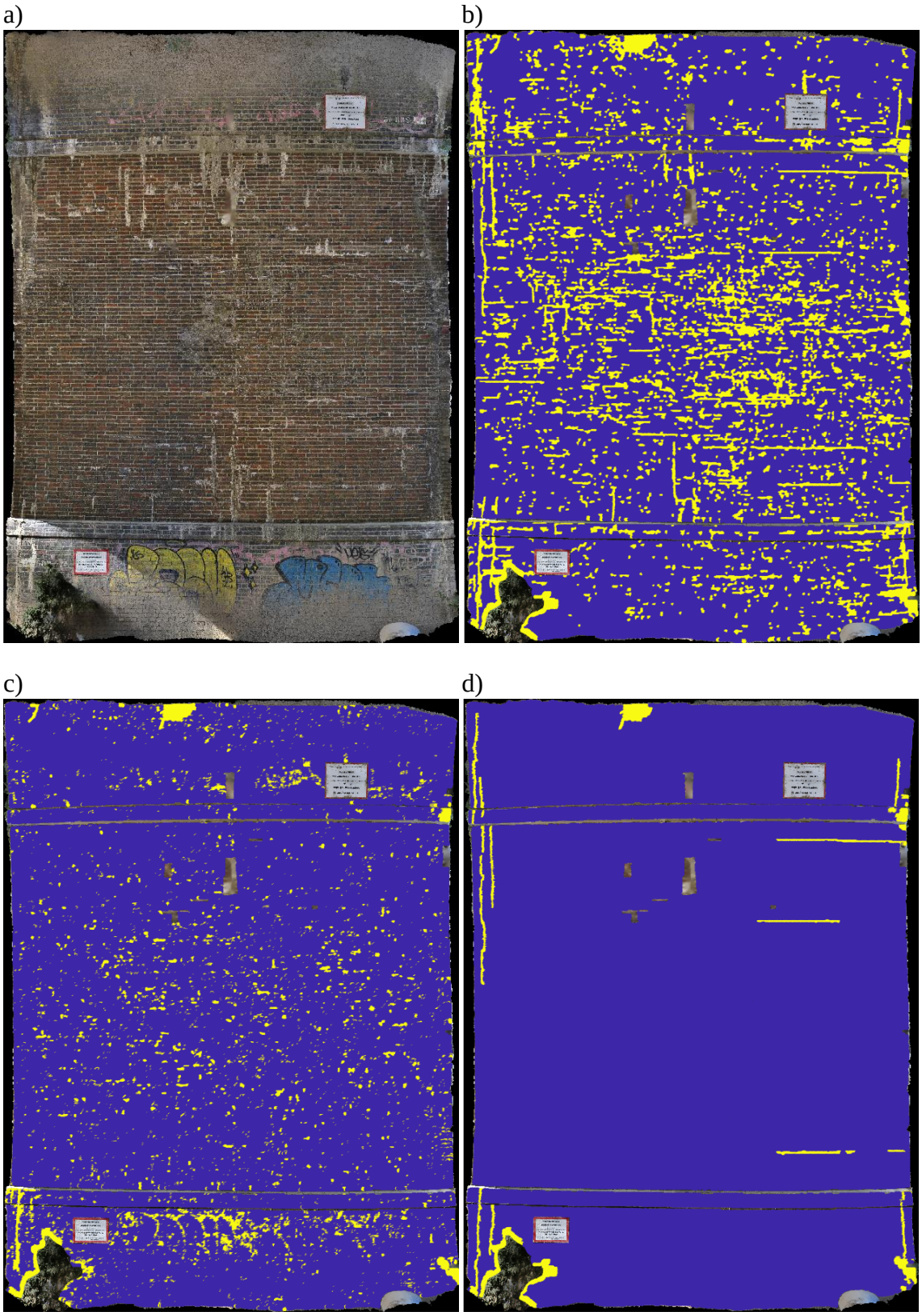
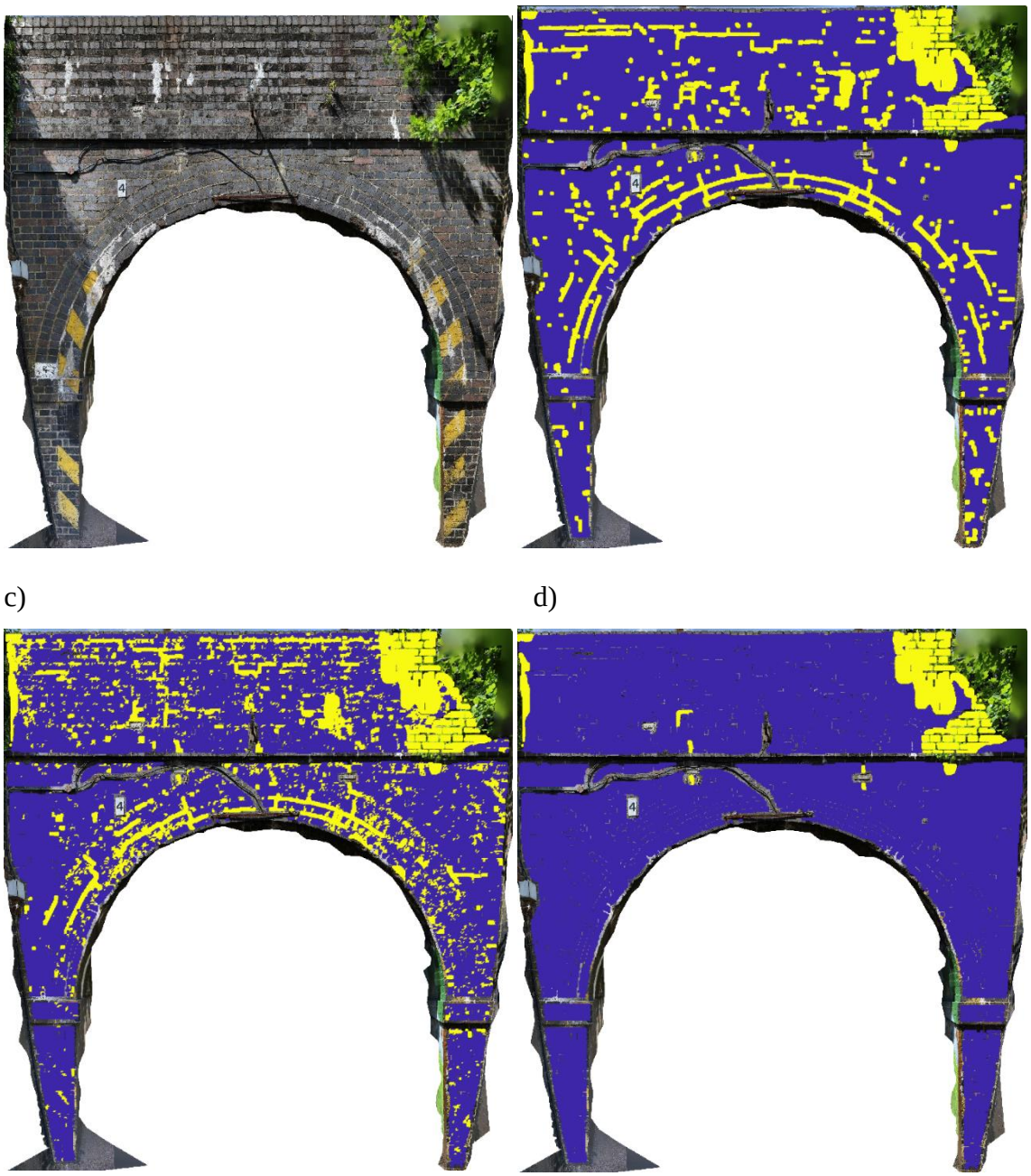


Figure 7-5: Defect classification on arch barrel and piers of bridge SBR12 showing: a) raw image, b) ground truth, c) automated defect classification predicted with 80% certainty, d) manual inspection

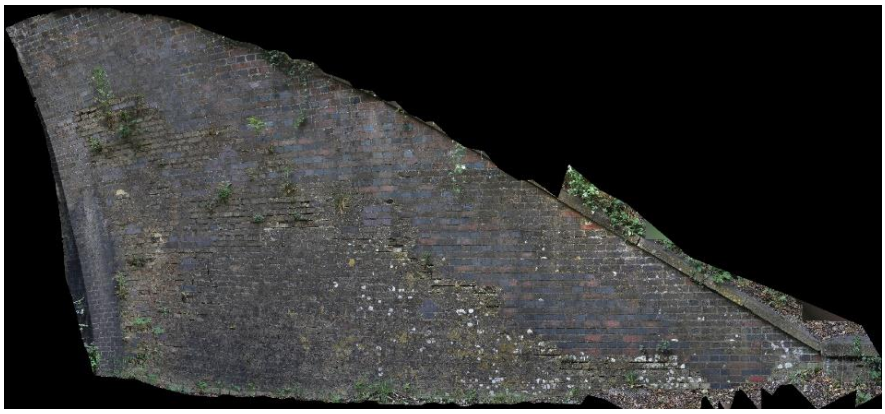
a) b)



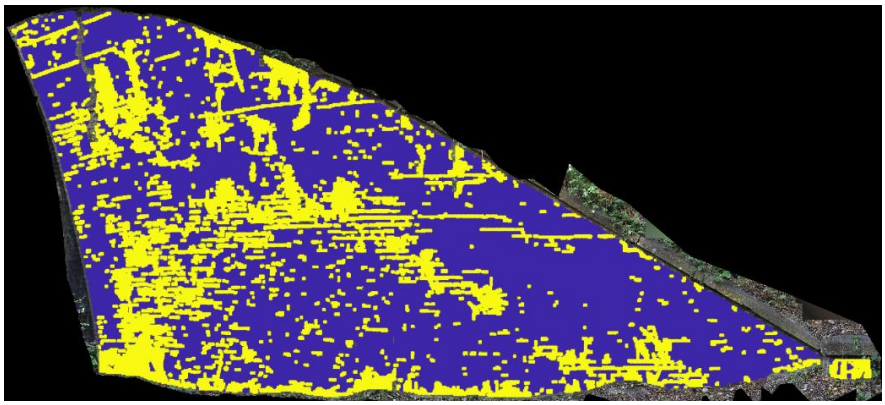
c) d)

Figure 7-6: Defect classification on front face of SBR4 showing: a) raw image, b) ground truth, c) automated defect classification predicted with 80% certainty, d) manual inspection

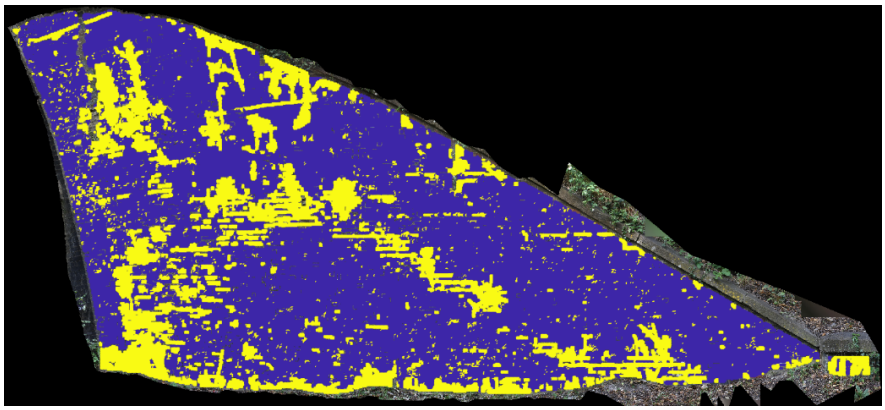
a)



b)



c)



d)

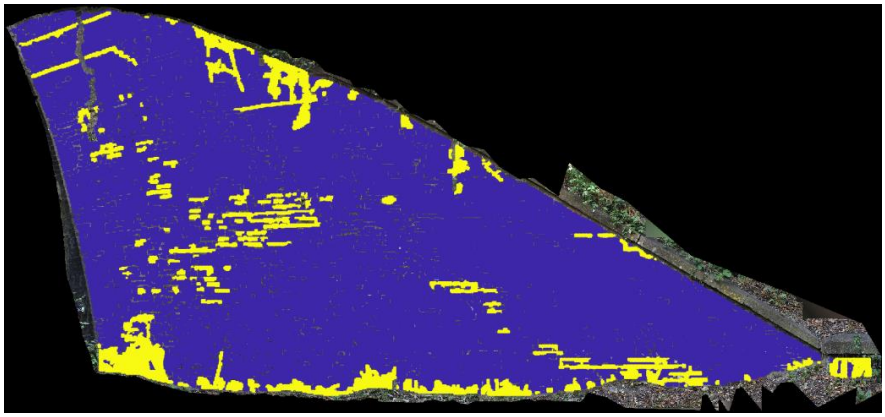
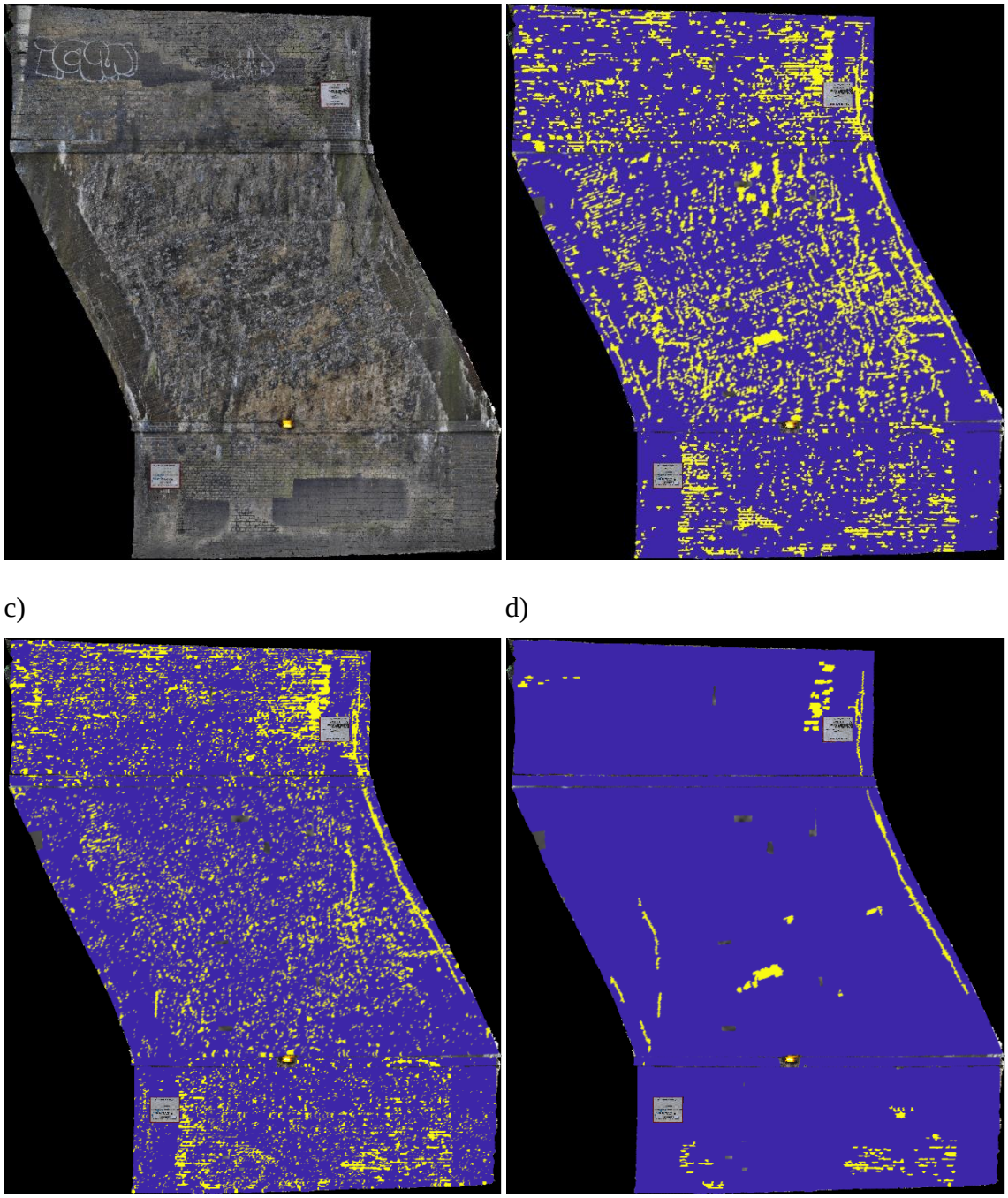


Figure 7-7: Defect classification on wing-wall section of SBR8 showing: a) raw image, b) ground truth, c) automated defect classification predicted with 80% certainty, d) manual inspection

a) b)



c) d)

Figure 7-8: Defect classification on arch barrel and piers of bridge SBR8 showing: a) raw image, b) ground truth, c) automated defect classification predicted with 80% certainty, d) manual inspection

7.3 Chapter summary

This chapter has sought to compare the performance of the developed automated classifier to that of manual inspection, as the existing methodology for assessing the condition of masonry arch bridges. A dataset of three bridges has been developed, whereby all visible surfaces have been imaged and annotated with data of defects comprising both the ground truth, and those identified during manual inspection. Manual inspection data was determined from the most recent detailed inspection report from each bridge. During this process it was observed that for two of the three bridges examined, the inspection reports showed photographs of only about half of the identified defects. The three bridge inspections were conducted by different inspectors suggesting significant variability in the inspection process and thoroughness.

The automated classifier used was from classification strategy 1 in Chapter 6, which was trained with no mortar and brick segmentation. This was used to classify image windows into defective and clean categories, and the resulting classifications were compared with both the ground truth and the manual inspection classifications. It was found that for the default threshold of 0.5, the automated classifier identified a far greater proportion of defects on the structure than were found during the manual inspection, but equally there were far more false positive detections. However, the ROC results showed that using a higher threshold of 0.95 lead to a similar performance between manual inspection and the automated classifier in terms of false positives, though still a larger proportion of defects were found with the automated classifier. Therefore, in general, the results suggest that the threshold could be set so that the performance of the automated classifier, as it stands, is at least as good as that of manual inspection. If it were desirable for a bridge inspector to track more minor defects, this would be possible with a lower threshold of detection. Additionally, the performance of the automated classifier also varied less between the different elements of bridges then manual inspection, suggesting human factors were responsible for some of the missed defects.

Finally, the outputs of the automated classifier and manual inspection were investigated visually for different images in the dataset. It was observed that the performance of the automated classifier deteriorated for conditions that were not that prevalent in the training dataset, such as with heavy surface deposits. It is therefore suggested that although the developed automated classifier already performs well, further improvements in its the performance could be achieved through expanding the training dataset to include more examples of such conditions. This is demonstrated in the performance of the classifier in

Chapter 6.1.2.2, on a dataset more like that trained upon, where its performance is far better.

8 CONCLUSIONS AND FUTURE WORK

8.1 Summary of main findings

The primary objective of this work as stated in Chapter 1.3. is as follows:

> *To evaluate and quantify the ability of different defect classification methodologies, to increase the understanding of the factors key to the performance of automated defect classification on masonry arch bridges. This therefore enables the development of a state-of-the-art automated image-based classifier with performance exceeding that of the existing manual inspection process.*

Working towards and fulfilling this objective, the findings are summarised as follows:

- The current standard industry practice for 2D visualising and processing of visual 3D models of masonry arch bridges is through orthophoto capture - the taking of photos of the model with the image plane parallel to the bridge surface. This achieves satisfactory results for most types of infrastructure. However, this work has demonstrated that for curved surfaces, such as the masonry arch, the resulting image suffers from distortion, with masonry towards the springing line of the arch having a lower image resolution than that at the crown. This has the potential to reduce the performance of the classification of defects as well as making measurement and quantification of defective areas difficult.

- For the mapping of mortar joints in noisy masonry surfaces, this work has compared the performance of a deterministic hardcoded method with two different deep learning classifiers, one giving an image-window based classification and the other a semantic classification. It has been shown that the semantic classifier has significantly outperformed the other approaches examined, as it is able to base its classification on both the localised texture of mortar and its interface as well as the expected mortar pattern.
- One of the key differences between masonry surfaces and concrete/asphalt surfaces, where most of the existing literature for defect detection focuses, is the mortar joint pattern. This pattern is visually more prominent than any defects, and a rudimentary classifier such as one based on edge detection would highlight mortar joints much more readily than defects. It has been shown that a much more capable classifier based on a Convolutional Neural Network is able to fully discern mortar joints when classifying defects, as testing with different strategies, both pre-segmenting mortar joints and not, showed a similar classification output. However, it is found that defect classification in mortar joint regions is less robust than in brick regions when the masonry conditions are noisy.
- It is found that detailed manual inspection identifies a far smaller proportion of the defects on the structure than an automated classifier achieves, and those defects which are identified are often neither photographed nor accurately position mapped on the structure. However, an automated classifier does result in many more false-positive classifications than are seen with manual inspection, though this can be improved through thresholding the classification confidence.

8.2 Primary contributions

To achieve the findings discussed in Chapter 8.1, existing analysis methods were improved and often for the first time applied to the context of masonry arch bridges. The most important new research contributions are as listed below:

- A mapping has been produced linking the underlying condition a masonry arch bridge is suffering from to the combination of visible defects that would be used to diagnose it. Additionally, the interventions needed to cure the bridge of its underlying condition and alleviate the symptoms of the visible defects have also been mapped. Such a mapping can be used for an end-to-end automation of bridge

condition diagnosis and management when combined with an automated defect detection tool.

- When developing a model to solve a classification problem, it is well known that a comprehensive dataset for training is one of the most important components for reliable performance. This work has developed a dataset of both image and geometry data for masonry arch bridges that is believed to be unequalled in both scale and breadth. Much of the dataset has also been annotated pixel-wise with different defect classes, mortar joints, and defects identified during manual inspection. This dataset has the potential to become a useful research tool for the future development of improved models for defect classification in all masonry surfaces, including masonry arch bridges.
- A novel methodology has been developed for the creation of ortho-rectified image textures of the curved arch barrel. Utilising photogrammetry to create a visual 3D model of the bridge, it has been demonstrated that the image texture of the arch barrel can be unwrapped from the 3D model using UV mapping. Such a technique results in a 2D image texture of the arch barrel surface which has minimal scaling of the masonry, especially when compared to the existing industry practice of orthophoto capture.
- A mortar joint detection algorithm has been developed that has been shown to accurately segment mortar joint regions from the background masonry. This has been tested on noisy images of masonry arch bridges, depicting surface features such as vegetation, graffiti, block painting, efflorescence, shadowing, and a coating of slurry, where it has been shown to have excellent performance. No previous method identified in the literature has been developed to cope with such levels of noise, which are commonplace on masonry arch bridges.
- An automated classifier has been developed for classifying image windows of masonry surfaces into those containing defects and those not. This has been tested across three different bridges, with a large quantity of noisy distractors, where its performance has been shown to be equal or better than the previous manual detailed inspection that had been conducted. Additionally, the classifier has also been tested on a dataset more like its training dataset in which its performance was far better, suggesting that with more training data, there is a possibility of significant performance improvement above that currently achieved by manual inspection.

Together, these contributions can lead to a significant performance improvement to the existing manual inspection process for masonry arch bridges.

8.3 Suggestions for future research

The presented work involved the development and application of automated defect detection tools, contributing towards better assessment of the condition of masonry arch bridges. The work also motivates numerous further areas of continued research:

- When tested on a dataset more like its training dataset, the developed defect detection classifier performed far better. The existing training dataset is made up of images primarily from the flat surfaces of masonry viaducts. These images contain far less noisy features than seen in the testing dataset of the complete surface of three individual bridges. This is partly because a good proportion of noisy features are experienced on the arch barrels of bridges. These surfaces do not get cleaned by rain making them susceptible to significant build-up of surface deposits. Additionally, the viaducts visited are generally in better condition than the three bridges used in the testing dataset, potentially due to being larger structures, further from public roads, and having had more recent maintenance. Optimum classification performance is achieved when the conditions in testing are better or equal to those in training. Therefore, future work could involve augmenting the existing training dataset with images containing worse masonry conditions. This could be achieved by splitting the existing testing dataset and additionally training on two of the bridges and testing on the third, or by collecting data from more bridges.
- The existing defect detection classifier segmented the masonry surface into defective and non-defective classes as a binary classification. Additional work would be required to train a multi-class classifier to classify all the separate defect classes individually if this were desired. The training and testing datasets have been annotated with each defect class separately, to aid such a future step.
- A potential route to reduce false positive defect classifications is to focus on the larger, more structurally significant defects. These defects are mostly easier to discern from the background masonry, so are less likely to be confused for image noise. This was examined by removing the smaller defects from the testing data, where it was shown that these defects were more likely to be predicted correctly.

A classifier trained to only classify the larger defects, could also result in reduced false positive detections.

- For mortar joint detection, both an image-window based, and a semantic based neural network model were tested, and the semantic based model was found to produce superior results, as classification could be based on local mortar and interface texture as well as the more global mortar pattern. Class Activation Mapping has shown that for defect detection, the image-window based classifier has learnt to locate defects semi-semantically within image windows despite being trained on window class, rather than semantic data. The currently developed defect detection methodology, using image windows means that each image window is classified individually based on its own merits. Using a semantic model for detecting defects in masonry could therefore result in an improvement in detection performance as both localised and more global information can be used for prediction.
- The existing work looks to locate defects within 2D masonry image textures. Future work could integrate this output back onto a 3D model of the bridge, so that defects could be spatially related. By using the developed mapping between underlying bridge conditions and visible defects, this 3D defect model could be used to automatically diagnose faults and solutions for bridges.

All these areas of future research directly relate to the contributions herein. As such, they indicate a path which demonstrates how the presented work could be extended and practically implemented to facilitate the future automation of the visual inspection of masonry arch bridges.

9 REFERENCES

A162 Barkston Road Underline Bridge. n.d. The ABC Railway Guide. Accessed September 17, 2021. https://abcrailwayguide.uk/bridge-cfm51-barkston-ash-north-yorkshire#.YUN0_51KiUk.

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. www.tensorflow.org.

Abdel-Qader I, Abudayyeh O, and Kelly ME. (2003). Analysis of Edge-Detection Techniques for Crack Identification in Bridges. *Journal of Computing in Civil Engineering* 17 (4): 255–63. https://doi.org/10.1061/(asce)0887-3801(2003)17:4(255).

Abdelkader EM, Moselhi O, Marzouk M, and Zayed T. (2020). Hybrid Elman Neural Network and an Invasive Weed Optimization Method for Bridge Defect Recognition. *Transportation Research Record: Journal of the Transportation Research Board*, November, 036119812096794. https://doi.org/10.1177/0361198120967943.

Agisoft. (2019). AgiSoft Metashape Standard (Version 1.5.3). 2019. https://www.agisoft.com/downloads/installer/.

Agnisarman S, Lopes S, Chalil Madathil K, Piratla K, and Gramopadhye A. (2019). A Survey of Automation-Enabled Human-in-the-Loop Systems for Infrastructure Visual Inspection. *Automation in Construction*. https://doi.org/10.1016/j.autcon.2018.10.019.

Ali L, Khan W, and Chaiyasarn K. (2019). DAMAGE DETECTION AND LOCALIZATION IN MASONRY STRUCTURE USING FASTER REGION CONVOLUTIONAL NETWORKS. *International Journal of GEOMATE* 17 (59): 98–105. https://doi.org/10.21660/2019.59.8272.

Alipour M and Harris DK. (2020). Increasing the Robustness of Material-Specific Deep Learning Models for Crack Detection across Different Materials. *Engineering Structures* 206 (March): 110157. https://doi.org/10.1016/j.engstruct.2019.110157.

Alipour M, Harris DK, and Miller GR. (2019). Robust Pixel-Level Crack Detection Using Deep Fully Convolutional Neural Networks. *Journal of Computing in Civil Engineering* 33 (6): 04019040. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000854.

Amhaz R, Chambon S, Idier J, and Baltazart V. (2016). Automatic Crack Detection on Two-Dimensional Pavement Images: An Algorithm Based on Minimal Path Selection. *IEEE Transactions on Intelligent Transportation Systems* 17 (10): 2718–29. https://doi.org/10.1109/TITS.2015.2477675.

Anand R, Mehrotra KG, Mohan CK, and Ranka S. (1993). An Improved Algorithm for Neural Network Classification of Imbalanced Training Sets. *IEEE Transactions on Neural Networks* 4 (6): 962–69. https://doi.org/10.1109/72.286891.

Armesto-González J, Riveiro-Rodríguez B, González-Aguilera D, and Rivas-Brea MT. (2010). Terrestrial Laser Scanning Intensity Data Applied to Damage Detection for Historical Buildings. *Journal of Archaeological Science* 37 (12): 3037–47. https://doi.org/10.1016/j.jas.2010.06.031.

Atkins. (2009). Review of Bridge Inspection Competence and Training Project Report : Final. http://www.bridgeforum.org/bof/projects/bict/Bridge Inspector Training and Competence Phase 1 Report Final.pdf.

Audley End Viaduct. n.d. The ABC Railway Guide. Accessed September 16, 2021. https://abcrailwayguide.uk/bridge-bgk15161-wendens-ambo-essex#.YUHlzJ1KiUk.

Autodesk. (2017). Drone Photogrammetry - Introducing Autodesk® ReCap™ Photo. *Autodesk*. https://blogs.autodesk.com/recap/introducing-recap-photo/.

Bennetts J, Webb G, Denton S, Vardanega PJ, and Loudon N. (2018). Quantifying Uncertainty in Visual Inspection Data. In *Maintenance, Safety, Risk, Management*

and Life-Cycle Performance of Bridges - Proceedings of the 9th International Conference on Bridge Maintenance, Safety and Management, IABMAS 2018, 2252–59. CRC Press/Balkema. https://doi.org/10.1201/9781315189390-306.

Bianco S, Cadene R, Celona L, and Napoletano P. (2018). Benchmark Analysis of Representative Deep Neural Network Architectures, October. https://doi.org/10.1109/ACCESS.2018.2877890.

Blender Online Community. (2018). Blender - a 3D Modelling and Rendering Package. Blender Foundation. http://www.blender.org.

Brodu N and Lague D. (2012). 3D Terrestrial Lidar Data Classification of Complex Natural Scenes Using a Multi-Scale Dimensionality Criterion: Applications in Geomorphology. *ISPRS Journal of Photogrammetry and Remote Sensing* 68 (1): 121–34. https://doi.org/10.1016/j.isprsjprs.2012.01.006.

Campbell K, Lydon M, Stevens N-A, Neeson T, and Taylor S. (2020). A Review of the Data Held on 3,437 Masonry Arch Bridges in Northern Ireland. *Civil Engineering Research Ireland*. Slovak Scientific Society for Physical Education and Sport and FIEP. https://doi.org/10.2/JQUERY.MIN.JS.

Canziani A, Paszke A, and Culurciello E. (2017). An Analysis of Deep Neural Network Models for Practical Applications. *2017 IEEE International Symposium on Circuits & Systems*. http://arxiv.org/abs/1605.07678.

Cavalagli N, Cluni F, and Gusella V. (2013). Evaluation of a Statistically Equivalent Periodic Unit Cell for a Quasi-Periodic Masonry. *International Journal of Solids and Structures* 50 (25–26): 4226–40. https://doi.org/10.1016/j.ijsolstr.2013.08.027.

cdcseacave. (2019). OpenMVS. http://cdcseacave.github.io/openMVS/.

Cha Y-J, Choi W, and Büyüköztürk O. (2017). Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Computer-Aided Civil and Infrastructure Engineering* 32 (5): 361–78. https://doi.org/10.1111/mice.12263.

Chaiyasarn K, Khan W, Ali L, Sharma M, Brackenbury D, and Dejong M. (2018). Crack Detection in Masonry Structures Using Convolutional Neural Networks and Support Vector Machines. In *35th International Symposium on Automation and Robotics in Construction (ISARC 2018) Crack*. https://www.iaarc.org/publications/fulltext/ISARC2018-Paper030.pdf.

Chaiyasarn K, Sharma M, Ali L, Khan W, and Poovarodom N. (2018). CRACK

DETECTION IN HISTORICAL STRUCTURES BASED ON CONVOLUTIONAL NEURAL NETWORK. *International Journal of GEOMATE* 15 (51): 240–51. https://doi.org/10.21660/2018.51.35376.

Chen S, Laefer DF, Asce M, Mangina ; Eleni, Iman Zolanvari ; S M, and Byrne J. (2019). UAV Bridge Inspection through Evaluated 3D Reconstructions. https://doi.org/10.1061/(ASCE).

Choi W and Cha YJ. (2020). SDDNet: Real-Time Crack Segmentation. *IEEE Transactions on Industrial Electronics* 67 (9): 8016–25. https://doi.org/10.1109/TIE.2019.2945265.

Chollet F and Others. (2015). Keras: The Python Deep Learning API. *Keras: The Python Deep Learning API*. https://keras.io/.

Cluni F, Costarelli D, Minotti AM, and Vinti G. (2015). Enhancement of Thermographic Images as Tool for Structural Analysis in Earthquake Engineering. *NDT and E International* 70 (March): 60–72. https://doi.org/10.1016/j.ndteint.2014.10.001.

Dais D, Bal İE, Smyrou E, and Sarhosis V. (2021). Automatic Crack Classification and Segmentation on Masonry Surfaces Using Convolutional Neural Networks and Transfer Learning. *Automation in Construction* 125 (May): 103606. https://doi.org/10.1016/j.autcon.2021.103606.

Davies C and Dennis S. (2021). The Importance of Structures Examinations: Why ORR Is Concerned by Network Rail Plans. *New Civil Engineer*, July 21, 2021. https://www.newcivilengineer.com/latest/the-importance-of-structures-examinations-why-orr-is-concerned-by-network-rail-plans-21-07-2021/.

Deng J, Dong W, Socher R, Li L-J, Kai Li, and Li Fei-Fei. (2009). ImageNet: A Large-Scale Hierarchical Image Database, 248–55. https://doi.org/10.1109/cvprw.2009.5206848.

Dorafshan S, Thomas RJ, Coopmans C, and Maguire M. (2018). Deep Learning Neural Networks for SUAS-Assisted Structural Inspections: Feasibility and Application. In *2018 International Conference on Unmanned Aircraft Systems, ICUAS 2018*, 874–82. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICUAS.2018.8453409.

Dorafshan S, Thomas RJ, and Maguire M. (2018). Comparison of Deep Convolutional Neural Networks and Edge Detectors for Image-Based Crack Detection in Concrete.

Construction and Building Materials. https://doi.org/10.1016/j.conbuildmat.2018.08.011.

East West Rail. (2021). MAKING MEANINGFUL CONNECTIONS. https://eastwestrail.co.uk/the-project/project-overview.

Ecm1/161 Viaduct. n.d. The ABC Railway Guide. Accessed September 17, 2021. https://abcrailwayguide.uk/bridge-ecm11611-wood-walton-cambridgeshire#.YUNxV51KiUk.

Ellenberg A, Kontsos A, Bartoli I, and Pradhan A. (2014). Masonry Crack Detection Application of an Unmanned Aerial Vehicle. *Computing In Civil and Building Engineering*, 1788–95. https://doi.org/10.1061/9780784413616.222.

Faro. n.d. FARO® SCENE Software | Software | FARO. Accessed September 20, 2021. https://www.faro.com/en/Products/Software/SCENE-Software.

FireAlpaca. (2021). FireAlpaca | Free Digital Painting Software. 2021. https://firealpaca.com/.

Forgács T, Sarhosis V, and Bagi K. (2017). Minimum Thickness of Semi-Circular Skewed Masonry Arches. *Engineering Structures* 140 (June): 317–36. https://doi.org/10.1016/j.engstruct.2017.02.036.

Forster A, Valero E, Bosché F, Hyslop E, Wilson L, and Turmel A. (2020). Digital Detectives. *RICS Built Environment* 17 (5): 22–24. https://doi.org/10.1111/1740-9713.01439.

Fuhrmann S, Langguth F, and Goesele M. (2014). MVE-A Multi-View Reconstruction Environment. *EUROGRAPHICS Workshops on Graphics and Cultural Heritage*. http://diglib.eg.org/.

Fujita Y and Hamamoto Y. (2010). A Robust Automatic Crack Detection Method from Noisy Concrete Surfaces. *Machine Vision and Applications* 22: 245–54. https://doi.org/10.1007/s00138-009-0244-5.

Galantucci RA and Fatiguso F. (2019). Advanced Damage Detection Techniques in Historical Buildings Using Digital Photogrammetry and 3D Surface Anlysis. *Journal of Cultural Heritage* 36 (March): 51–62. https://doi.org/10.1016/j.culher.2018.09.014.

Galantucci RA, Fatiguso F, and Galantucci LM. (2018). A Proposal for a New Standard

Quantification of Damages of Cultural Heritages, Based on 3D Scanning. *SCIRES-IT* 8 (1): 121–38. https://doi.org/10.2423/i22394303v8n1p121.

Girardeau-Montaut D. (2014). CloudCompare. EDF R&D, Telecom ParisTech. https://cloudcompare.org/.

Gopalakrishnan K, Khaitan SK, Choudhary A, and Agrawal A. (2017). Deep Convolutional Neural Networks with Transfer Learning for Computer Vision-Based Data-Driven Pavement Distress Detection. *Construction and Building Materials* 157 (December): 322–30. https://doi.org/10.1016/J.CONBUILDMAT.2017.09.110.

Graybeal BA, Rolander DD, Phares BM, Moore ME, and Washer GA. (2001). Reliability and Accuracy of In-Depth Inspection of Highway Bridges. *Transportation Research Record: Journal of the Transportation Research Board* 1749 (1): 93–99. https://doi.org/10.3141/1749-14.

Grimm CT. (1988). Masonry Cracks: A Review of the Literature. In *Masonry: Materials, Design, Construction, and Maintenance*, 257–80. Philadelphia. https://www.masonryinfo.org/wp-content/uploads/2018/07/Masonry-Cracks-A-Review-of-the-Literature-Grimm.pdf.

Griwodz C, Gasparini S, Calvet L, Gurdjos P, Castan F, Maujean B, Lanthony Y, and Lillo G de. (2021). AliceVision Meshroom: An Open-Source 3D Reconstruction Pipeline. *Proceedings of the 12th ACM Multimedia Systems Conference*, September, 241–47. https://doi.org/10.1145/3458305.3478443.

Guo J, Wang Q, Li Y, and Liu P. (2020). Façade Defects Classification from Imbalanced Dataset Using Meta Learning-based Convolutional Neural Network. *Computer-Aided Civil and Infrastructure Engineering* 35 (12): 1403–18. https://doi.org/10.1111/mice.12578.

Hallee MJ, Napolitano RK, Reinhart WF, and Glisic B. (2021). Crack Detection in Images of Masonry Using Cnns. *Sensors* 21 (14). https://doi.org/10.3390/S21144929.

Hallermann N and Morgenthal G. (2016). From Aerial Photography to 3-Dimensional Inspection of Bridges. In *IABSE Conference, Guangzhou 2016: Bridges and Structures Sustainability - Seeking Intelligent Solutions - Report*, 546–53. International Association for Bridge and Structural Engineering (IABSE). https://doi.org/10.2749/222137816819258898.

Hassan MU, Akcamete-Gungor A, and Meral C. (2017). Investigation of Terrestrial Laser Scanning Reflectance Intensity and RGB Distributions to Assist Construction Material Identification. In , 507–15. https://doi.org/10.24928/jc3-2017/0312.

He K, Zhang X, Ren S, and Sun J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:770–78. https://doi.org/10.1109/CVPR.2016.90.

Helmerich R, Bien J, and Buhr B. (2007). Guideliene for Inspection and Comndition Assessment of Existing European Railway Bridges. *Sustainable Bridges*, 1–259.

Hertford Viaduct. n.d. The ABC Railway Guide. Accessed September 17, 2021. https://abcrailwayguide.uk/bridge-hdb391-hertford-hertfordshire#.YUNvaZ1KiUk.

Hinton G. (2012). Neural Networks for Machine Learning. In *Human and Machine Hearing*, 419–40. https://doi.org/10.1017/9781139051699.031.

Historic England. (1967). Chappel Viaduct- 1238766. National Heritage List for England. 1967. https://historicengland.org.uk/listing/the-list/list-entry/1238766.

———. (1980). WELWYN RAILWAY VIADUCT- 1348122. National Heritage List for England. 1980. https://historicengland.org.uk/listing/the-list/list-entry/1348122.

———. (1987). WELLAND VIADUCT, Harringworth - 1264288. National Heritage List for England. 1987. https://historicengland.org.uk/listing/the-list/list-entry/1264288.

Hough P. (1960). METHOD AND MEANS FOR RECOGNIZING COMPLEX PATTERNS, issued March 25, 1960. http://www.google.com/patents/US3069654?printsec=description#v=onepage&q&f=false.

Hui L, Park M, and Brilakis I. (2014). Automated In-Place Brick Counting for Facade Construction Progress Estimation. *Computing in Civil and Building Engineering*, 958–65. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000423.

Hüthwohl P and Brilakis I. (2018). Detecting Healthy Concrete Surfaces. *Advanced Engineering Informatics* 37 (August): 150–62. https://doi.org/10.1016/j.aei.2018.05.004.

Ibrahim Y, Nagy B, and Benedek C. (2019). Cnn-Based Watershed Marker Extraction

for Brick Segmentation in Masonry Walls. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11662 LNCS:332–44. Springer Verlag. https://doi.org/10.1007/978-3-030-27202-9_30.

Jahanshahi MR, Masri SF, Padgett CW, and Sukhatme GS. (2013). An Innovative Methodology for Detection and Quantification of Cracks through Incorporation of Depth Perception. *Machine Vision and Applications* 24 (2): 227–41. https://doi.org/10.1007/s00138-011-0394-0.

Kassotakis N, Sarhosis V, Peppa M V., and Mills J. (2021). Quantifying the Effect of Geometric Uncertainty on the Structural Behaviour of Arches Developed from Direct Measurement and Structure-from-Motion (SfM) Photogrammetry. *Engineering Structures* 230 (March): 111710. https://doi.org/10.1016/j.engstruct.2020.111710.

Koch C, Georgieva K, Kasireddy V, Akinci B, and 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.

Laefer DF, Gannon J, and Deely E. (2010). Reliability of Crack Detection Methods for Baseline Condition Assessments. *Journal of Infrastructure Systems* 16 (2): 129–37. https://doi.org/10.1061/(ASCE)1076-0342(2010)16:2(129).

Larbi JA. (2004). Microscopy Applied to the Diagnosis of the Deterioration of Brick Masonry. *Construction and Building Materials* 18 (5): 299–307. https://doi.org/10.1016/j.conbuildmat.2004.02.002.

Laurent J, Fox-Ivey R, Sanchez Dominguez F, Antonio J, and Garcia R. (2014). Use of 3D Scanning Technology for Automated Inspection of Tunnels. In *The World Tunnel Congress*, 1–10.

Leica Geosystems. (2014). Leica Pegasus : Two Mobile Mapping Solution.

Li Y, Zhao W, Zhang X, and Zhou Q. (2018). A Two-Stage Crack Detection Method for Concrete Bridges Using Convolutional Neural Networks. *IEICE Transactions on Information and Systems* E101D (12): 3249–52. https://doi.org/10.1587/transinf.2018EDL8150.

Liu Y, Yao J, Lu X, Xie R, and Li L. (2019). DeepCrack: A Deep Hierarchical Feature

Learning Architecture for Crack Segmentation. *Neurocomputing* 338 (April): 139–53. https://doi.org/10.1016/j.neucom.2019.01.036.

Lourenço PB. (2002). Computations on Historic Masonry Structures. *Progress in Structural Engineering and Materials* 4 (3): 301–19. https://doi.org/10.1002/pse.120.

Loverdos D, Sarhosis V, Adamopoulos E, and Drougkas A. (2021). An Innovative Image Processing-Based Framework for the Numerical Modelling of Cracked Masonry Structures. *Automation in Construction* 125 (May): 103633. https://doi.org/10.1016/j.autcon.2021.103633.

Marin B, Brown K, and Erden MS. (2021). Automated Masonry Crack Detection with Faster R-CNN. *IEEE International Conference on Automation Science and Engineering* 2021-Augus (August): 333–40. https://doi.org/10.1109/CASE49439.2021.9551683.

MathWorks. (2018). Matlab. The MathWorks Inc. https://uk.mathworks.com/products/matlab.html.

McCormick N and Lord J. (2012). Digital Image Correlation for Structural Measurements. *Proceedings of the Institution of Civil Engineers - Civil Engineering* 165 (4): 185–90. https://doi.org/10.1680/cien.11.00040.

McCormick NJ, Kimkeran SA, Najimi A, and Jonas D. (2014). Assessing the Condition of Railway Assets Using DIFCAM: Results from Tunnel Examinations. *6th IET Conference on Railway Condition Monitoring (RCM 2014)*. https://doi.org/10.1049/cp.2014.1002.

McKibbins L, Melbourne C, Sawar N, and Gaillard C. (2006). *Masonry Arch Bridges: Condition Appraisal and Remedial Treatment (C656)*. CIRIA, London, UK.

McRobbie S. (2008). Automated Inspection of Highway Structures. *Automated Inspection of Highway*, 1–31. https://trl.co.uk/sites/default/files/PPR412.V3.pdf.

McRobbie SG. (2015). Development of an Image Based System for Routine Visual Inspection of UK Highways Bridges. University of Birmingham. https://etheses.bham.ac.uk//id/eprint/6184/1/McRobbie15PhD_redacted.pdf.

Megaw ED. (1979). Factors Affecting Visual Inspection Accuracy. *Applied Ergonomics* 10 (1): 27–32. https://doi.org/10.1016/0003-6870(79)90006-1.

Meroño JE, Perea AJ, Aguilera MJ, and Laguna AM. (2015). Recognition of Materials and Damage on Historical Buildings Using Digital Image Classification. *South African Journal of Science* 111 (1–2): 1–9. https://doi.org/10.17159/sajs.2015/20140001.

Mett M and Eder S. (2019). 3D Tunnel Inspection with Photogrammetric and Hybrid Systems. *Shotcrete for Underground Support XIV*, November. https://dc.engconfintl.org/shotcrete_xiv/10.

Microsoft. (2017). Image Composite Editor - Microsoft Research. *MICROSOFT*. https://www.microsoft.com/en-us/research/product/computational-photography-applications/image-composite-editor/.

Moore M, Phares B, Graybeal B, Rolander D, and Washer G. (2001). Reliability of Visual Inspection for Highway Bridges.

Napolitano R and Glisic B. (2019). Methodology for Diagnosing Crack Patterns in Masonry Structures Using Photogrammetry and Distinct Element Modeling. *Engineering Structures* 181 (February): 519–28. https://doi.org/10.1016/j.engstruct.2018.12.036.

Network Rail. (2015). National Rail Infrastructure Limited Annual Return 2015.

———. (2017). *NR/L3/CIV/006. Handbook for the Examination of Structures*. UK.

Nguyen TS, Avila M, and Begot S. (2009). Automatic Detection and Classification of Defect on Road Pavement Using Anisotropy Measure. In *European Signal Processing Conference*, 617–21. https://hal.archives-ouvertes.fr/hal-00666919.

Oakley Viaduct. n.d. The ABC Railway Guide. Accessed September 17, 2021. https://abcrailwayguide.uk/bridge-spc2481-oakley-bedford#.YUNv4p1KiUk.

Office of Rail and Road. (2007). Annual Assessment of Network Rail 2006-07, 41. http://www.orr.gov.uk/__data/assets/pdf_file/0012/2037/339.pdf.

———. (2017). Network Rail Monitor. http://www.orr.gov.uk/__data/assets/pdf_file/0015/25206/network-rail-monitor-2016-17-q3-4.pdf.

Oliveira H and Correia PL. (2013). Automatic Road Crack Detection and Characterization. *IEEE Transactions on Intelligent Transportation Systems* 14 (1): 155–68. https://doi.org/10.1109/TITS.2012.2208630.

OpenSource. (2020). Hugin - Panorama Photo Stitcher. *Sourceforg.Net*. http://hugin.sourceforge.net.

Orbán Z. (2004). ASSESSMENT, RELIABILITY AND MAINTENANCE OF MASONRY ARCH RAILWAY BRIDGES IN EUROPE. In *Arch Bridges IV - Advances in Assessment, Structural Design and Construction*, edited by P Roca and C Molins, 152–61. http://cn.arch-bridges.fzu.edu.cn/attach/2016/10/30/244079.pdf.

Orbán Z and Gutermann M. (2009). Assessment of Masonry Arch Railway Bridges Using Non-Destructive in-Situ Testing Methods. *Engineering Structures* 31 (10): 2287–98. https://doi.org/10.1016/j.engstruct.2009.04.008.

Oses N, Dornaika F, and Moujahid A. (2014). Image-Based Delineation and Classification of Built Heritage Masonry. *Remote Sensing* 6 (3): 1863–89. https://doi.org/10.3390/rs6031863.

Otsu N. (1979). THRESHOLD SELECTION METHOD FROM GRAY-LEVEL HISTOGRAMS. *IEEE Trans Syst Man Cybern* SMC-9 (1): 62–66. https://doi.org/10.1109/tsmc.1979.4310076.

Özgenel ÇF and Sorguç AG. (2018). Performance Comparison of Pretrained Convolutional Neural Networks on Crack Detection in Buildings. In *35th International Symposium on Automation and Robotics in Construction (ISARC 2018)*.

Parker C. (2013). Digital Imaging For Condition Asset Management. *RailEngineer*, 2013.

Perez H, Tah JHM, and Mosavi A. (2019). Deep Learning for Detecting Building Defects Using Convolutional Neural Networks. *Sensors (Switzerland)* 19 (16). https://doi.org/10.3390/s19163556.

Peterborough Viaduct. n.d. The ABC Railway Guide. Accessed September 17, 2021. https://abcrailwayguide.uk/bridge-ecm11841-fletton-and-woodston-peterborough#.YUNwSp1KiUk.

Phares BM, Washer GA, Rolander DD, Graybeal BA, and Moore M. (2004). Routine Highway Bridge Inspection Condition Documentation Accuracy and Reliability. *Journal of Bridge Engineering* 9 (4): 403–13. https://doi.org/10.1061/ASCE1084-070220049:4403.

Pitchford A, Gaillard C, Melbourne C, and Barnes A. (2003). CONDITION APPRAISAL AND REMEDIAL TREATMENT MASONRY AND BRICK ARCH BRIDGES.

http://www.bridgeforum.org/brf/brf1/p02_carlos_sicilia_gaillard.pdf.

Pozo S Del, Herrero-Pascual J, Felipe-García B, Hernández-López D, Rodríguez-Gonzálvez P, and González-Aguilera D. (2016). Multispectral Radiometric Analysis of Façades to Detect Pathologies from Active and Passive Remote Sensing. *Remote Sensing* 8 (1): 80. https://doi.org/10.3390/rs8010080.

Prasanna P, Dana KJ, Gucunski N, Basily BB, La HM, Lim RS, and Parvardeh H. (2016). Automated Crack Detection on Concrete Bridges. *IEEE Transactions on Automation Science and Engineering* 13 (2): 591–99. https://doi.org/10.1109/TASE.2014.2354314.

Quagliarini E, Clini P, and Ripanti M. (2017). Fast, Low Cost and Safe Methodology for the Assessment of the State of Conservation of Historical Buildings from 3D Laser Scanning: The Case Study of Santa Maria in Portonovo (Italy). *Journal of Cultural Heritage* 24 (March): 175–83. https://doi.org/10.1016/j.culher.2016.10.006.

Riveiro B., Caamaño JC, Arias P, and Sanz E. (2011). Photogrammetric 3D Modelling and Mechanical Analysis of Masonry Arches: An Approach Based on a Discontinuous Model of Voussoirs. *Automation in Construction* 20 (4): 380–88. https://doi.org/10.1016/j.autcon.2010.11.008.

Riveiro Belén, Lourenço PB, Oliveira D V., González-Jorge H, and Arias P. (2016). Automatic Morphologic Analysis of Quasi-Periodic Masonry Walls from LiDAR. *Computer-Aided Civil and Infrastructure Engineering* 31 (4): 305–19. https://doi.org/10.1111/mice.12145.

River Can and Occupation Road Viaduct. n.d. The ABC Railway Guide. Accessed September 17, 2021. https://abcrailwayguide.uk/bridge-ltn11491-moulsham-and-central-essex#.YUNuTZ1KiUk.

Rose P, Aaron B, Tamir DE, Lu L, Hu J, and Shi H. (2014). Supervised Computer-Vision-Based Sensing of Concrete Bridges for Crack-Detection and Assessment. *Transportation Research Board 93rd Annual Meeting*, no. 14–3857. http://amonline.trb.org/trb-59976-2014-1.2467145/t-1110-1.2491552/299-1.2477069/14-3857-1.2492557/14-3857-1.2492558?qr=1.

Ruzin S. (2011). Nyquist Sampling Rate. Digital Imaging for Biologists. Springer-Verlag. 2011. https://microscopy.berkeley.edu/courses/dib/sections/02images/sampling.html.

Saiz-Jimenez C and Marszałek M. (2004). Deterioration of Stone in Some Monuments Exposed to Air Pollution: A Cracow Case Study. In *Air Pollution and Cultural Heritage*, 151–54. CRC Press. https://doi.org/10.1201/b17004-22.

Samy MP, Foong S, Soh GS, and Soh, Gim SonYeo KS. (2016). Automatic Optical & Laser-Based Defect Detection and Classification in Brick Masonry Walls. *2016 IEEE Region 10 Conference (TENCON)*, 3521–24.

Sánchez-Aparicio LJ, Pozo S Del, Ramos LF, Arce A, and Fernandes FM. (2018). Heritage Site Preservation with Combined Radiometric and Geometric Analysis of TLS Data. *Automation in Construction* 85 (January): 24–39. https://doi.org/10.1016/j.autcon.2017.09.023.

Sbr/4 Spring Rd Underline Bridge. n.d. The ABC Railway Guide. Accessed September 17, 2021. https://abcrailwayguide.uk/bridge-sbr41-letchworth-south-west-hertfordshire#.YUN1pJ1KiUk.

Sbr/8 Norton Road Underline Bridge. n.d. The ABC Railway Guide. Accessed September 17, 2021. https://abcrailwayguide.uk/bridge-sbr81-baldock-town-hertfordshire#.YUN1rZ1KiUk.

Sbr /12 Underline Bridge. n.d. The ABC Railway Guide. Accessed September 17, 2021. https://abcrailwayguide.uk/bridge-sbr121-bygrave-hertfordshire#.YUHl1p1KiUk.

Schmitt KM, Riddington JR, Young RCD, Budgett DM, and Chatwin CR. (2000). Image Processing Applied to Brick Quality Control. *International Journal of Advanced Manufacturing Technology* 16 (6): 434–40. https://doi.org/10.1007/s001700050175.

Schmugge SJ, Rice L, Nguyen NR, Lindberg J, Grizzi R, Joffe C, and Shin MC. (2016). Detection of Cracks in Nuclear Power Plant Using Spatial-Temporal Grouping of Local Patches. In *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/WACV.2016.7477601.

Schönberger J. (2020). COLMAP — COLMAP 3.7 Documentation. https://colmap.github.io/index.html.

Scotland's Oldest Bridges. (2021). Parts of a Bridge and Glossary. 2021. https://scotlandsoldestbridges.co.uk/parts-of-a-bridge-and-gloss.html.

Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, and Batra D. (2016). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization.

International Journal of Computer Vision 128 (2): 336–59. https://doi.org/10.1007/s11263-019-01228-7.

Sheehan JJ and Drury CG. (1971). The Analysis of Industrial Inspection. *Applied Ergonomics* 2 (2): 74–78. https://doi.org/10.1016/0003-6870(71)90073-1.

Shi Y, Cui L, Qi Z, Meng F, and Chen Z. (2016). Automatic Road Crack Detection Using Random Structured Forests. *IEEE Transactions on Intelligent Transportation Systems* 17 (12): 3434–45. https://doi.org/10.1109/TITS.2016.2552248.

Shin HC, Roth HR, Gao M, Lu L, Xu Z, Nogues I, Yao J, Mollura D, and Summers RM. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging* 35 (5): 1285–98. https://doi.org/10.1109/TMI.2016.2528162.

Sithole G. (2008). Detection of Bricks in a Masonry Wall. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 567–72.

Siwowski T. (2015). *Bridge Engineering Selected Issues*. Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszow, Poland. https://www.researchgate.net/publication/326132087_Bridge_Engineering_-_Selected_Issues.

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, and Wojna Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–26. IEEE. http://arxiv.org/abs/1512.00567.

Tung SH, Shih MH, and Sung WP. (2008). Development of Digital Image Correlation Method to Analyse Crack Variations of Masonry Wall. *Sadhana* 33 (6): 767–79. http://www.springerlink.com/content/0kkh763nkm5232n1/%5Cnpapers3://publication/uuid/3B2FF270-DBDA-486F-B89F-9DE9774FA2AA.

Valero E, Forster A, Bosché F, Hyslop E, Wilson L, and Turmel A. (2019). Automated Defect Detection and Classification in Ashlar Masonry Walls Using Machine Learning. *Automation in Construction*. https://pureapps2.hw.ac.uk/ws/portalfiles/portal/24882267/2019_AutoConML.pdf.

Valero Enrique, Bosché F, and Forster A. (2018). Automatic Segmentation of 3D Point Clouds of Rubble Masonry Walls, and Its Application to Building Surveying, Repair and Maintenance. *Automation in Construction* 96: 29–39.

https://doi.org/10.1016/j.autcon.2018.08.018.

Valero Enrique, Bosché F, Forster A, and Hyslop E. (2019). Historic Digital Survey: Reality Capture and Automatic Data Processing for the Interpretation and Analysis of Historic Architectural Rubble Masonry. In *Structural Analysis of Historical Constructions: An Interdisciplinary Approach*, 18:388–96. Springer International Publishing. https://doi.org/10.1007/978-3-319-99441-3_41.

Valero Enrique, Bosché F, Forster A, M'Beirick I, Wilson L, Turmel A, and Hyslop E. (2020). Development of a Novel Open Tool for the Segmentation of 3D Point Clouds of Masonry Walls. In *14th International Congress on the Deterioration and Conservation of Stone*.

Valero Enrique, Forster A, Bosché F, Renier C, Hyslop E, and Wilson L. (2018). High Level-of-Detail BIM and Machine Learning for Automated Masonry Wall Defect Surveying. In *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.

Vryniotis V. (2018). The Batch Normalisation Layer of Keras Is Broken. DatumBox. 2018. https://blog.datumbox.com/the-batch-normalization-layer-of-keras-is-broken/.

Wallbank EJ and Department of Transport. (1989). *The Performance of Concrete in Bridges: A Survey of 200 Highway Bridges*. H.M. Stationery Office. https://books.google.co.uk/books?id=jhFCAQAAIAAJ.

Wang N, Zhao Q, Li S, Zhao X, and Zhao P. (2018). Damage Classification for Masonry Historic Structures Using Convolutional Neural Networks Based on Still Images. *Computer-Aided Civil and Infrastructure Engineering* 33 (12): 1073–89. https://doi.org/10.1111/mice.12411.

Wang N, Zhao X, Zhao P, Zhang Y, Zou Z, and Ou J. (2019). Automatic Damage Detection of Historic Masonry Buildings Based on Mobile Deep Learning. *Automation in Construction* 103 (July): 53–66. https://doi.org/10.1016/j.autcon.2019.03.003.

Wu C. (2011). VisualSFM : A Visual Structure from Motion System. 2011. http://ccwu.me/vsfm/.

Yang X, Li H, Yu Y, Luo X, Huang T, and Yang X. (2018). Automatic Pixel-Level Crack

Detection and Measurement Using Fully Convolutional Network. *Computer-Aided Civil and Infrastructure Engineering* 33 (12): 1090–1109. https://doi.org/10.1111/mice.12412.

Ye C, Acikgoz S, Pendrigh S, Riley E, and DeJong MJ. (2018). Mapping Deformations and Inferring Movements of Masonry Arch Bridges Using Point Cloud Data. *Engineering Structures* 173 (October): 530–45. https://doi.org/10.1016/J.ENGSTRUCT.2018.06.094.

Yosinski J, Clune J, Bengio Y, and Lipson H. (2014). How Transferable Are Features in Deep Neural Networks? *Advances in Neural Information Processing Systems* 4 (January): 3320–28. http://arxiv.org/abs/1411.1792.

Yuan L, Guo J, and Wang Q. (2020). Automatic Classification of Common Building Materials from 3D Terrestrial Laser Scan Data. *Automation in Construction* 110 (February): 103017. https://doi.org/10.1016/j.autcon.2019.103017.

Zhang A, Wang KCP, Asce M, Fei ; Yue, Liu Y, Tao S, Chen ; Cheng, Li JQ, and Li B. (2018). Deep Learning-Based Fully Automated Pavement Crack Detection on 3D Asphalt Surfaces with an Improved CrackNet. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000775.

Zhang L, Yang F, Daniel Zhang Y, and Zhu YJ. (2016). Road Crack Detection Using Deep Convolutional Neural Network. In *2016 IEEE International Conference on Image Processing (ICIP)*, 3708–12. IEEE. https://doi.org/10.1109/ICIP.2016.7533052.

Zhang Y, Macorini L, and Izzuddin B. (2018). Numerical Investigation of Arches in Brick-Masonry Bridges. *Structure and Infrastructure Engineering* 14 (1): 14–32. https://doi.org/10.1080/15732479.2017.1324883.

Zhang Y, Macorini L, and Izzuddin BA. (2016). Mesoscale Partitioned Analysis of Brick-Masonry Arches. *Engineering Structures* 124 (October): 142–66. https://doi.org/10.1016/j.engstruct.2016.05.046.

Zhao H, Shi J, Qi X, Wang X, and Jia J. (2017). Pyramid Scene Parsing Network. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:6230–39. https://doi.org/10.1109/CVPR.2017.660.

Zhou B, Khosla A, Lapedriza A, Oliva A, and Torralba A. (2016). Learning Deep Features for Discriminative Localization. *Proceedings of the IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition 2016-Decem: 2921–29. https://doi.org/10.1109/CVPR.2016.319.

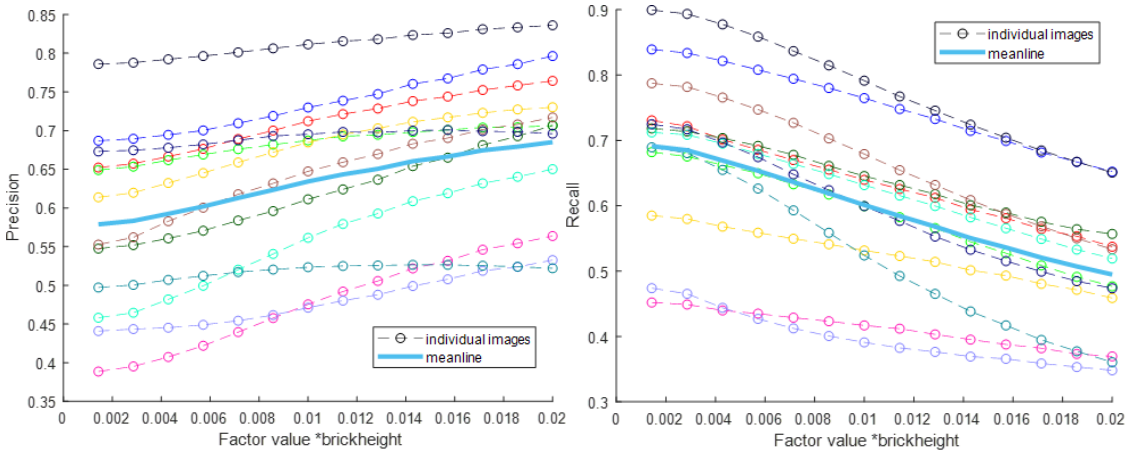
Zhou B, Zhao H, Puig X, Fidler S, Barriuso A, and Torralba A. (2017). Scene Parsing through ADE20K Dataset. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:5122–30. https://doi.org/10.1109/CVPR.2017.544.

10 APPENDICES

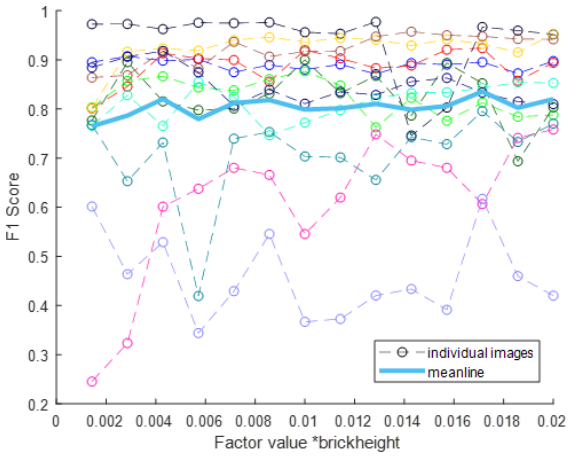
APPENDIX 1: DETERMINISTIC MORTAR JOINT PATTERN RECOGNITION FACTOR ANALYSIS

1: Standard deviation of gaussian filter used prior to edge detection

Edges detection performance:

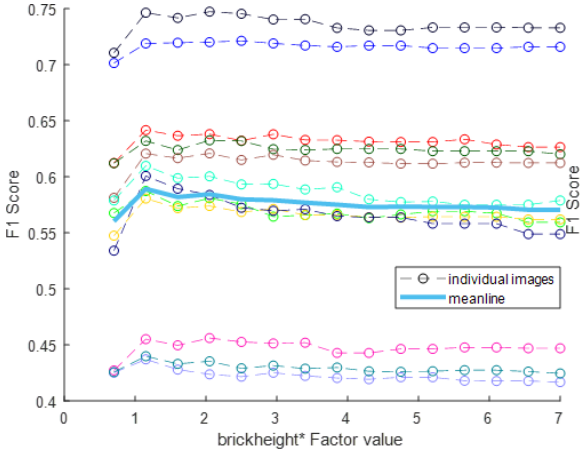


Mortar joint detection performance:

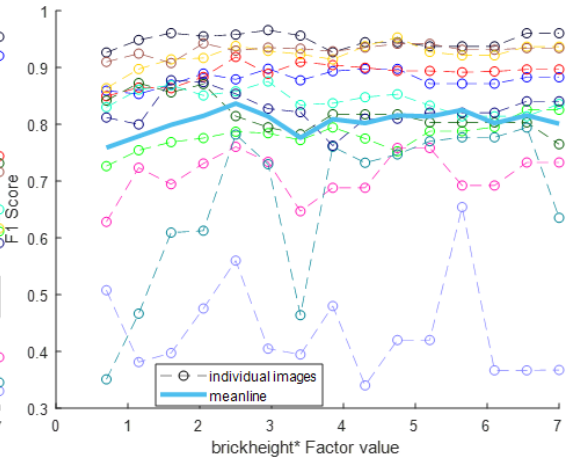


2: Size of regions image broken into for processing

Edges detection performance:

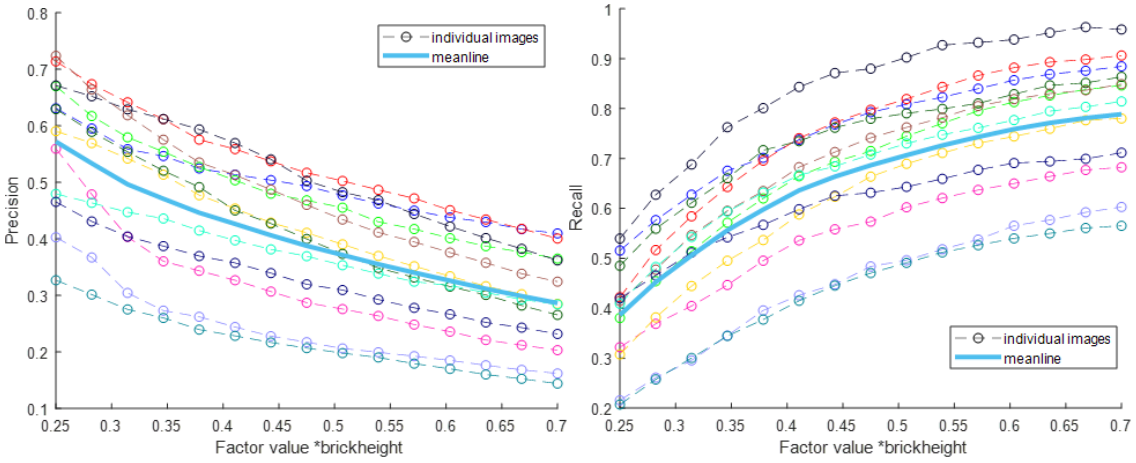


Mortar joint detection performance:

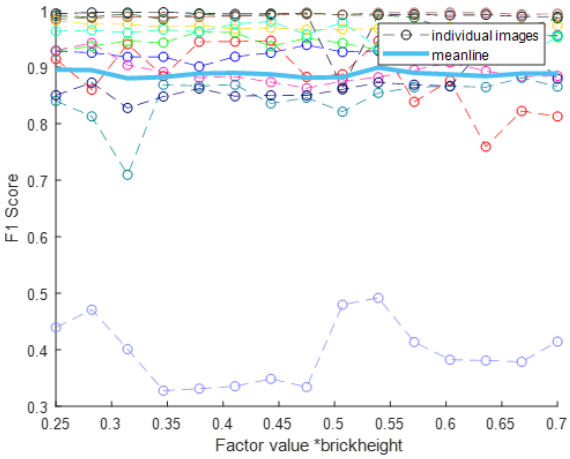


3: Fill gap used for horizontal Hough line detection, detecting the straight lines of horizontal mortar joints

Straight lines detection performance:

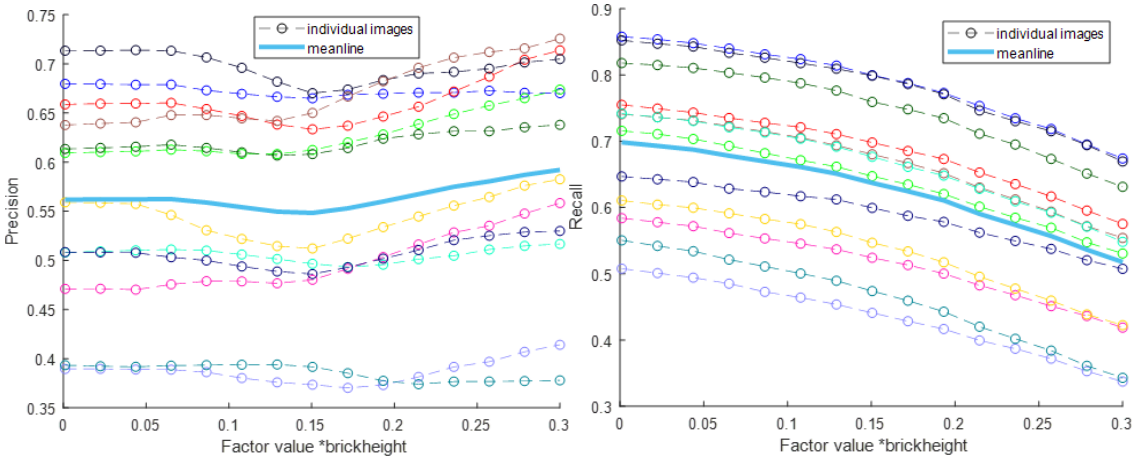


Horizontal mortar joint detection performance:

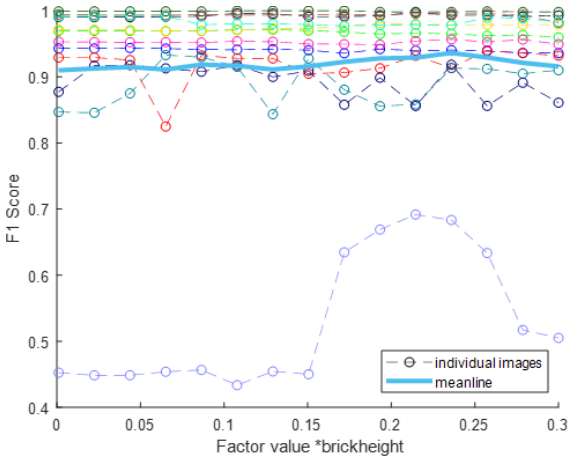


4: Minimum length for initial straight lines for Hough line detection of horizontal mortar joints

Straight lines detection performance:

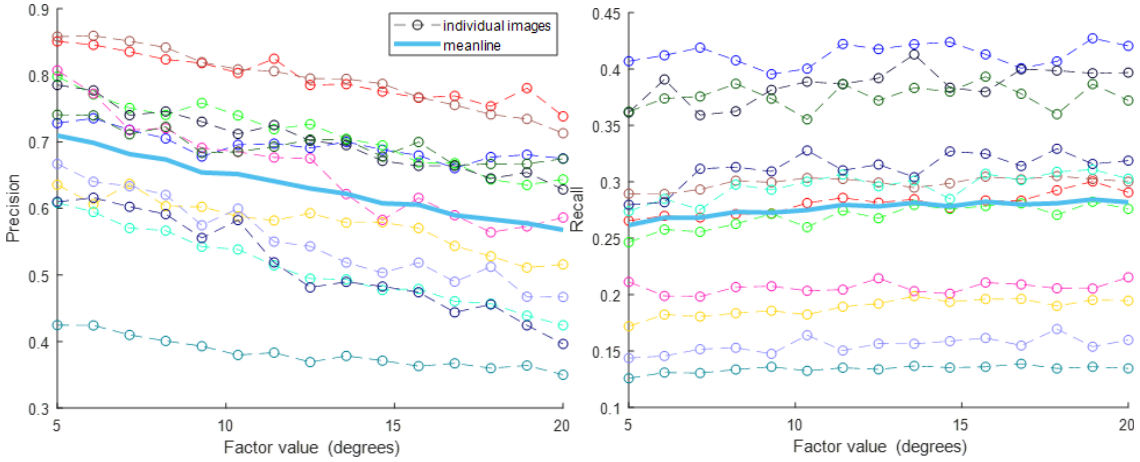


Horizontal mortar joint detection performance:

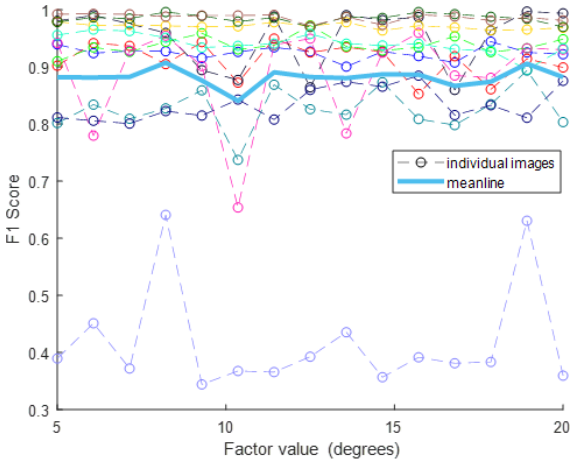


5: Search angle range for horizontal straight lines for Hough line detection of horizontal mortar joints

Straight lines detection performance:

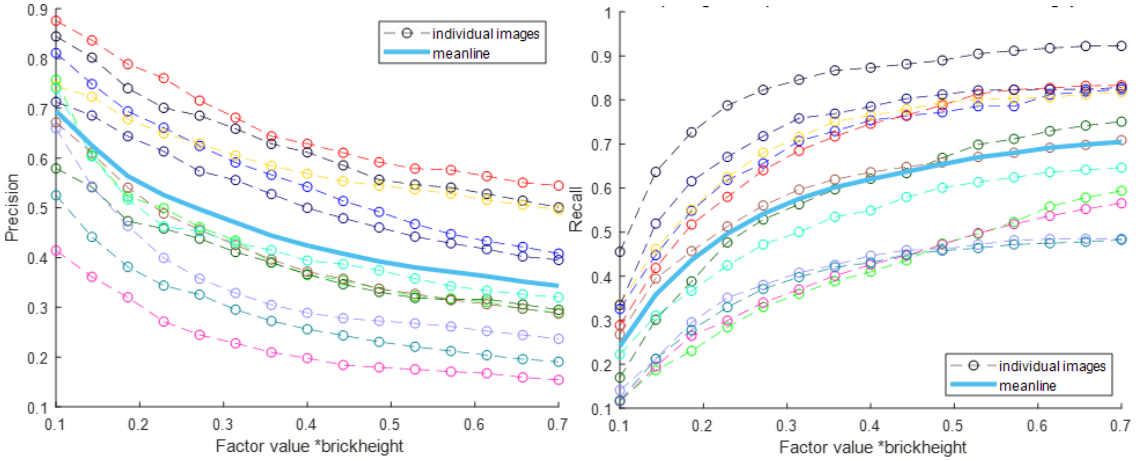


Horizontal mortar joint detection performance:

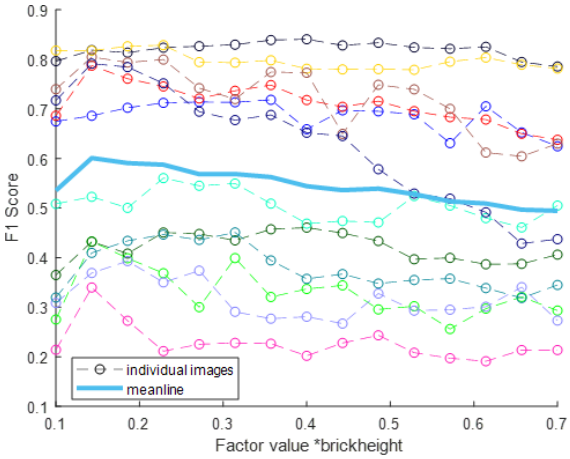


6: Fill gap used for vertical Hough line detection, detecting the straight lines of vertical mortar joints

Straight lines detection performance:

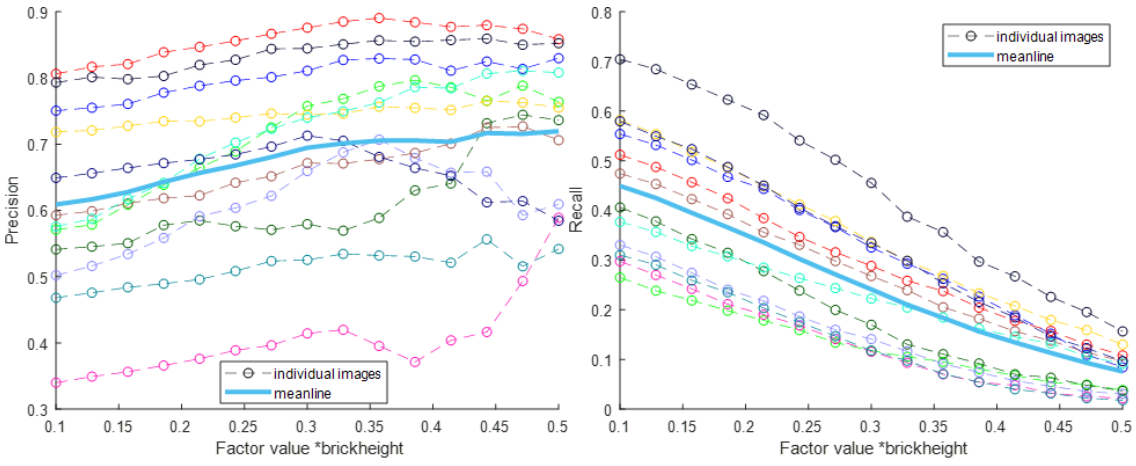


Vertical mortar joint detection performance:

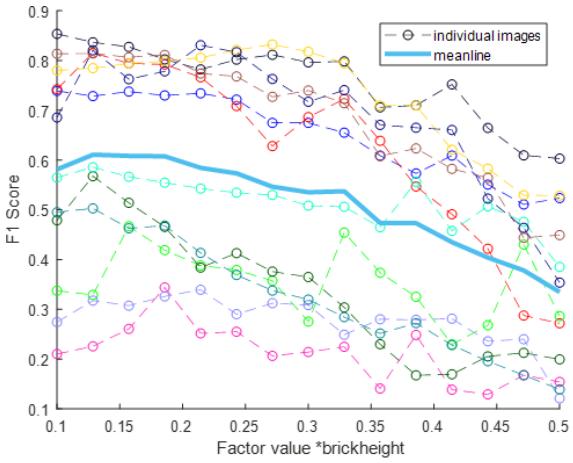


7: Minimum length for initial straight lines for Hough line detection of vertical mortar joints

Straight lines detection performance:

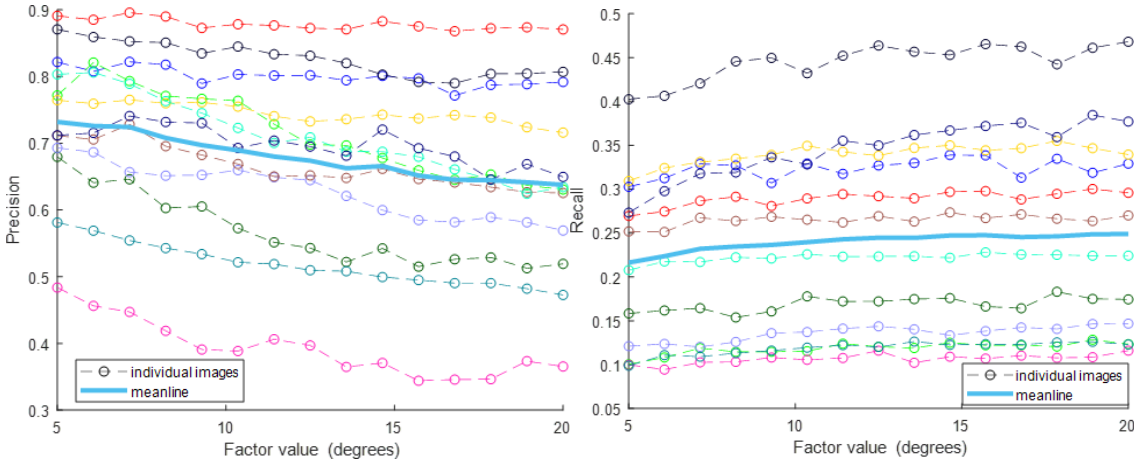


Vertical mortar joint detection performance:

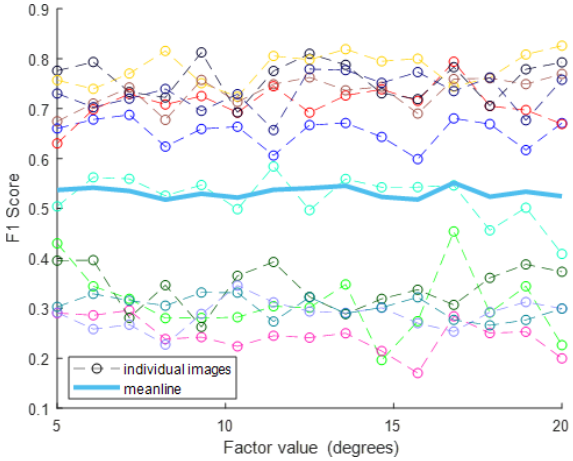


8: Search angle range for vertical straight lines for Hough line detection of vertical mortar joints

Straight lines detection performance:

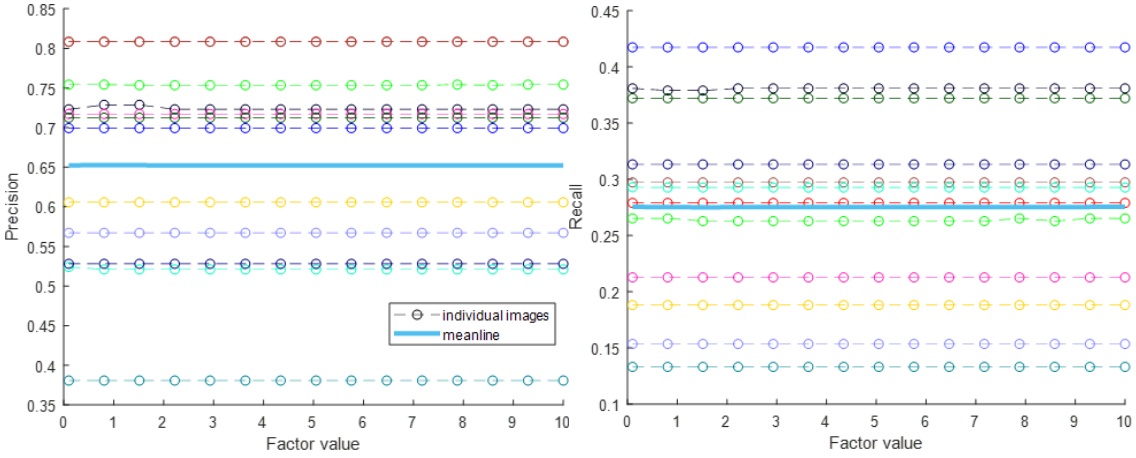


Vertical mortar joint detection performance:

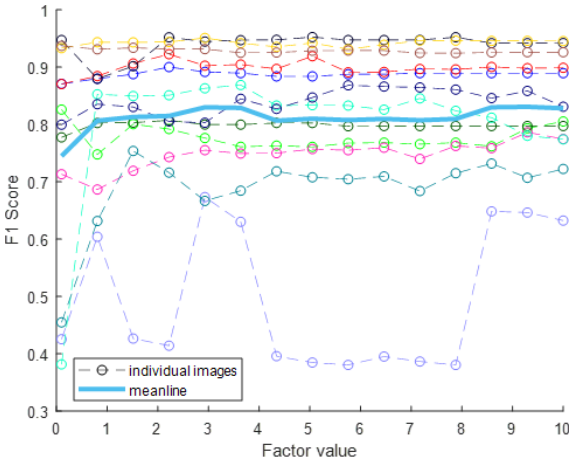


9: Bandwidth for fitting distribution to rotation angles detected for grouping similar angles

Straight lines detection performance:

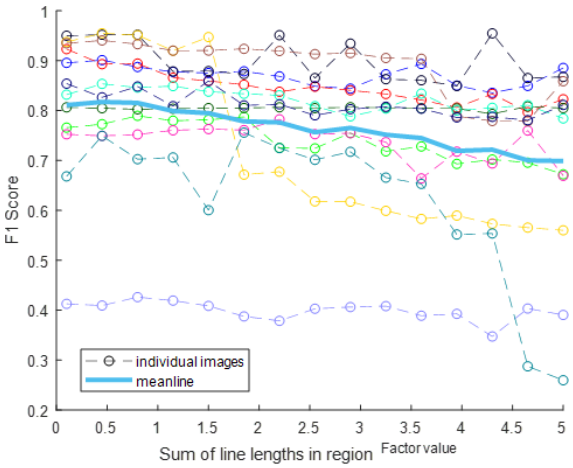


Mortar joint detection performance:



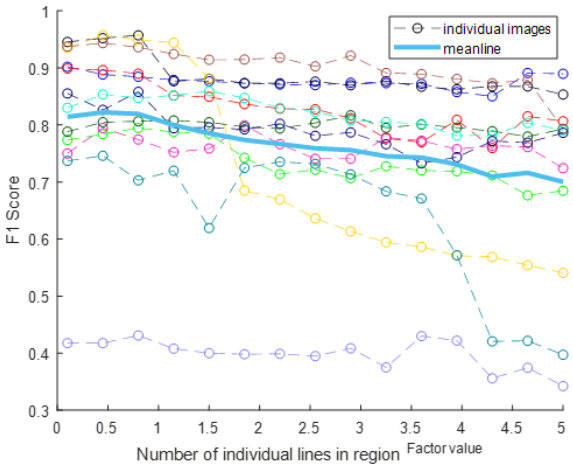
10: Region confidence determination: straight line length weighting

Mortar joint detection performance:



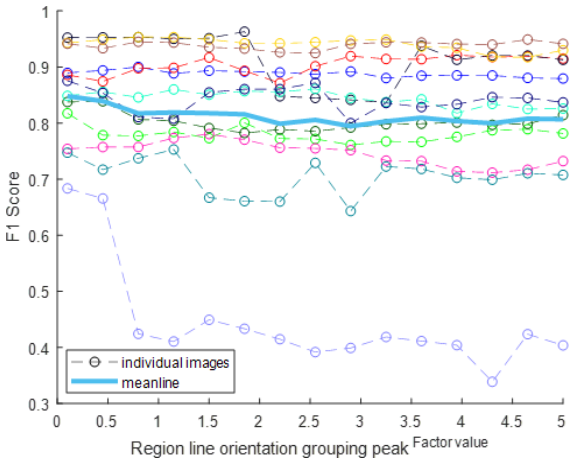
11: Region confidence determination: number of straight lines weighting

Mortar joint detection performance:



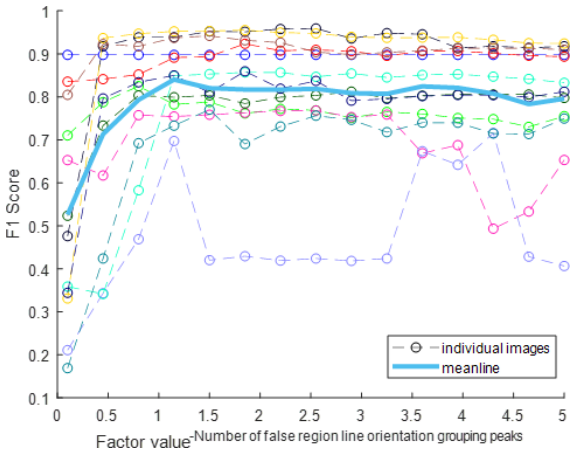
12: Region confidence determination: straight line angle consistency weighting

Mortar joint detection performance:



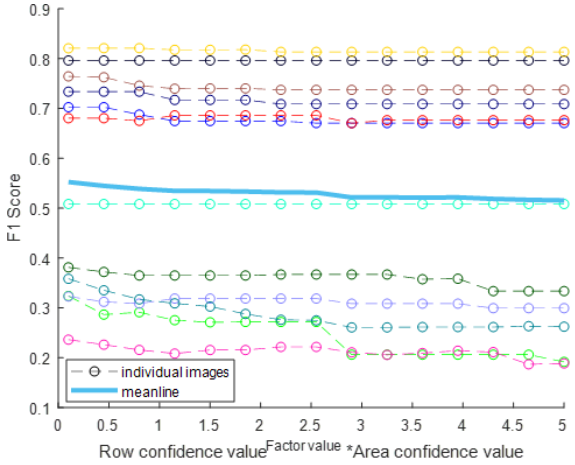
13: Region confidence determination: incorrect straight line angle weighting

Mortar joint detection performance:



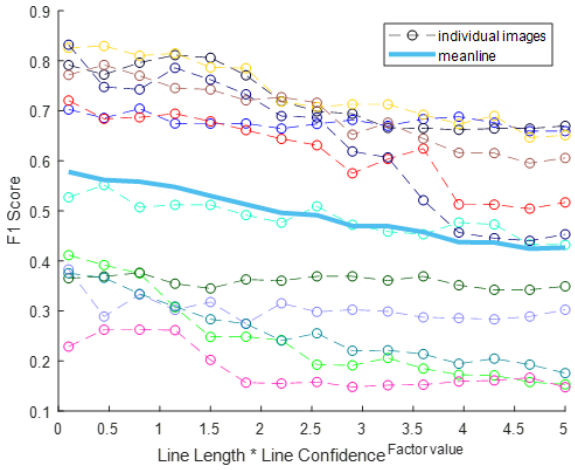
14: Vertical mortar joint confidence determination: relative weighting of effect of row confidence and area confidence

Vertical mortar joint detection performance:



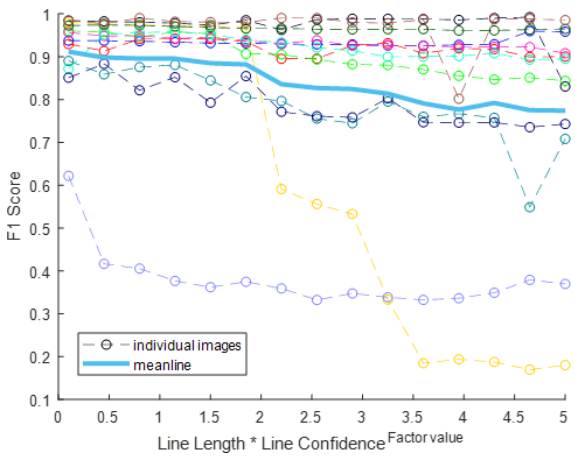
15: Vertical mortar joint effect of line confidence on length weighting

Vertical mortar joint detection performance:



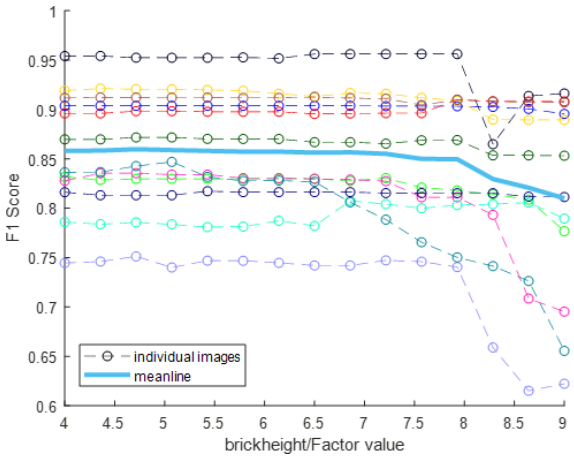
16: Horizontal mortar joint effect of line confidence on length weighting

Horizontal mortar joint detection performance:



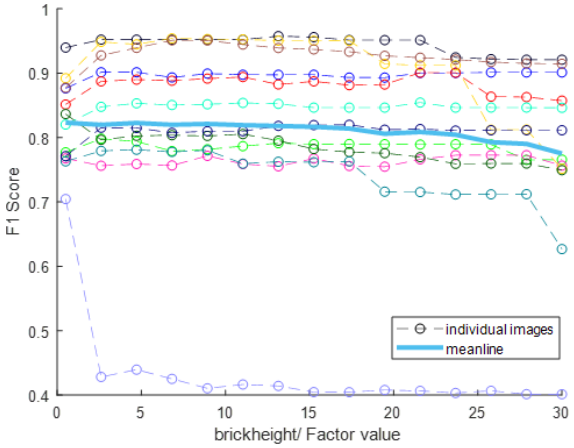
17: Bandwidth for fitting distribution to straight line locations for mortar joint grouping

Mortar joint detection performance:



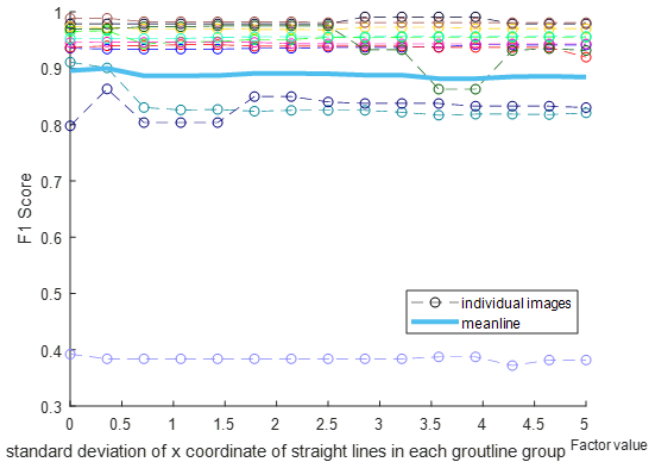
18: Bandwidth for fitting distribution to mortar joint spacing values

Mortar joint detection performance:



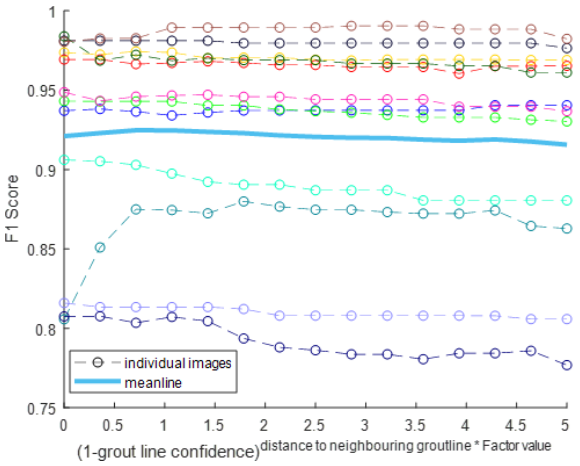
19: Horizontal mortar joint confidence determination: weighting of spread of straight lines detected across mortar joint (meaning less extrapolation)

Horizontal mortar joint detection performance:



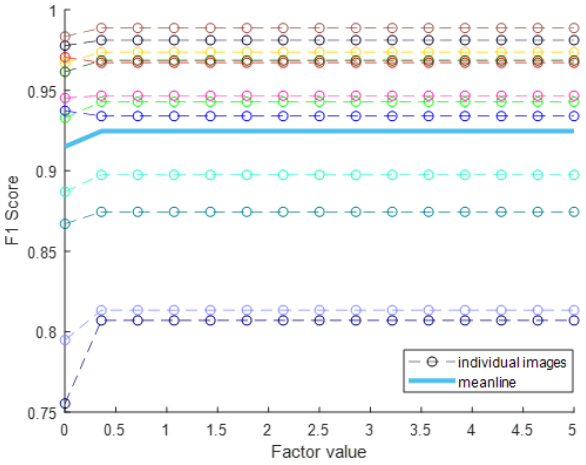
20: Horizontal mortar joint separation distance decay factor for gradient calculation

Horizontal mortar joint detection performance:



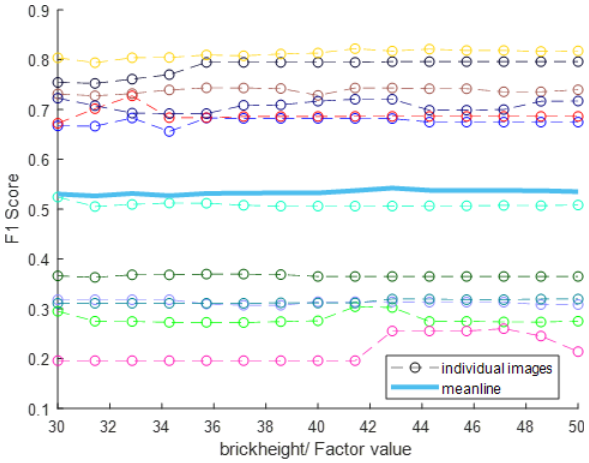
21: Bandwidth of fitted distribution grouping gradient values of horizontal mortar joints

Horizontal mortar joint detection performance:



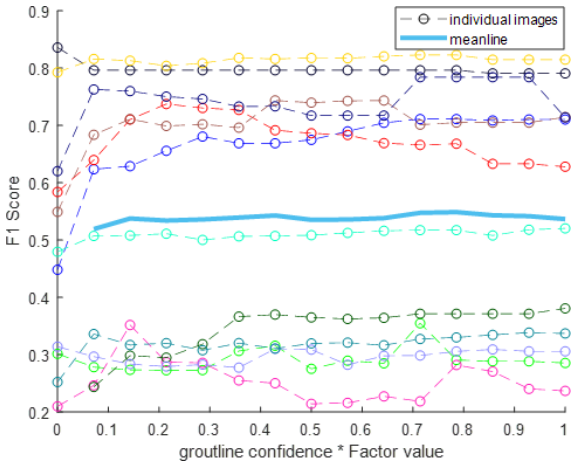
22: Bandwidth of fitted distribution detecting spread of vertical mortar joint points to determine doubly defined mortar joints

Vertical mortar joint detection performance:



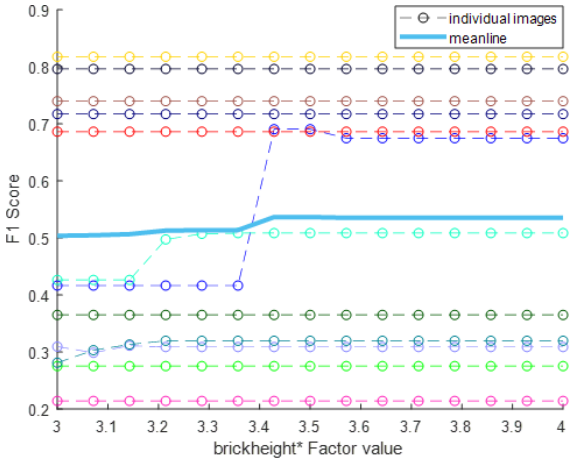
23: Confidence penalty for singly defined vertical mortar joints

Vertical mortar joint detection performance:



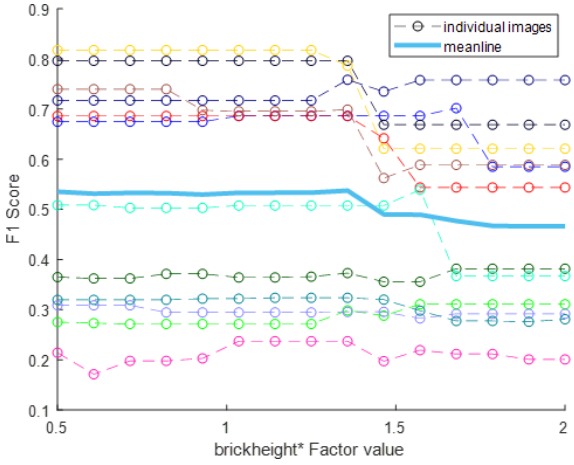
24: Threshold for maximum allowable vertical mortar joint spacing

Vertical mortar joint detection performance:



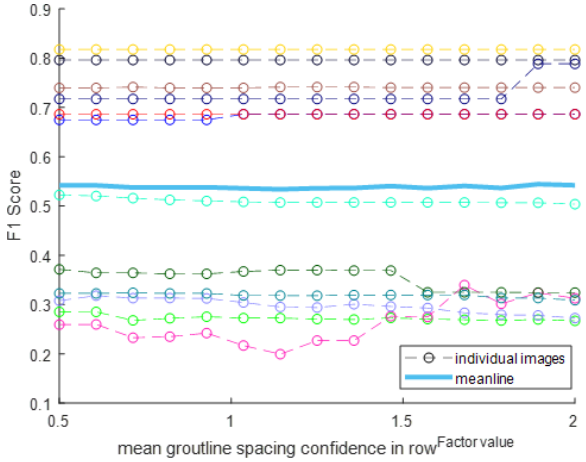
25: Threshold for minimum allowable vertical mortar joint spacing

Vertical mortar joint detection performance:



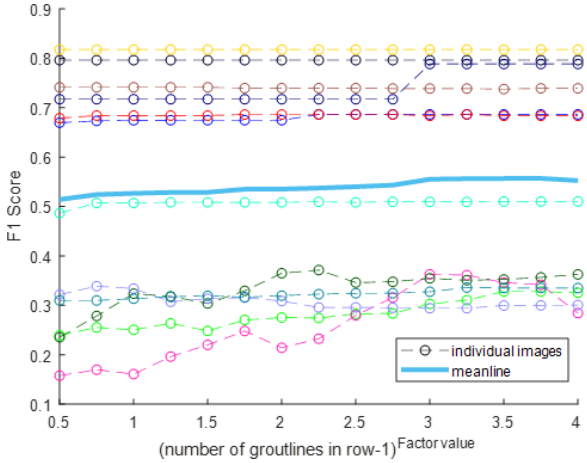
26: Brick course brick spacing confidence: individual mortar joint spacing confidence weighting

Vertical mortar joint detection performance:



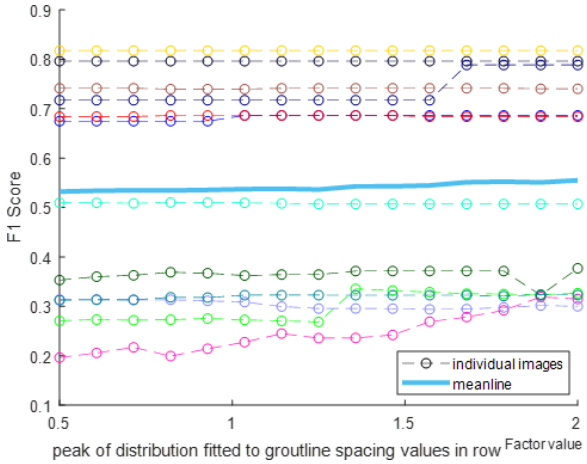
27: Brick course brick spacing confidence: number of mortar joints in row weighting

Vertical mortar joint detection performance:



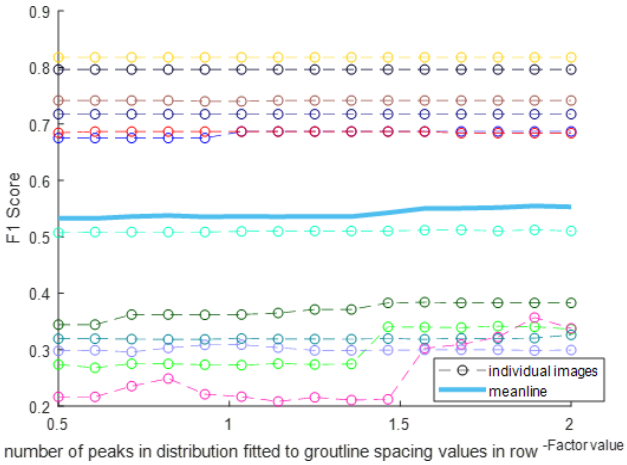
28: Brick course brick spacing confidence: consistency of spacing in row weighting

Vertical mortar joint detection performance:



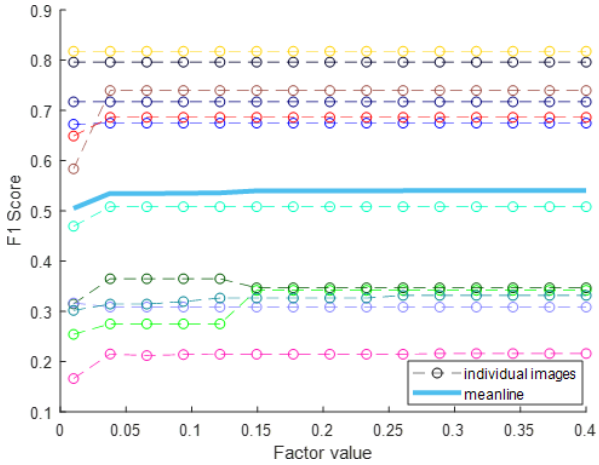
29: Brick course brick spacing confidence: penalisation for inconsistent spacing in row

Vertical mortar joint detection performance:



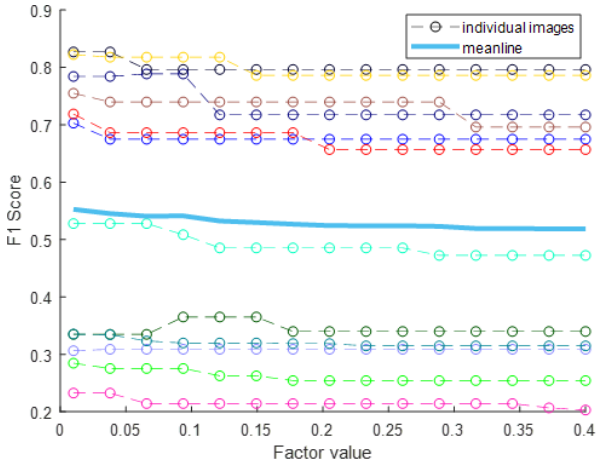
30: Vertical mortar joint spacing, confidence threshold for forcing pattern

Vertical mortar joint detection performance:



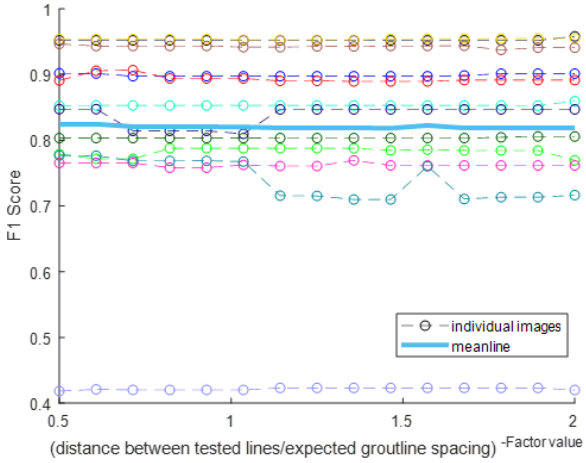
31: Vertical mortar joint spacing, proportional deviation from pattern threshold for forcing pattern

Vertical mortar joint detection performance:



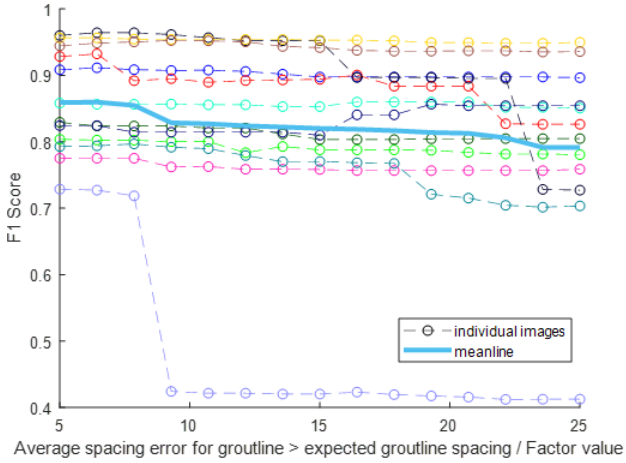
32: Incorrect mortar joint removal, rate of decay in confidence due to separation between tested lines

Mortar joint detection performance:



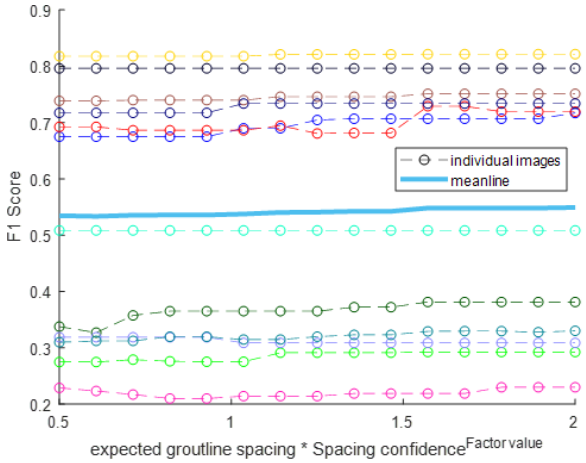
33: Incorrect mortar joint removal, threshold for spacing error required for removal

Mortar joint detection performance:



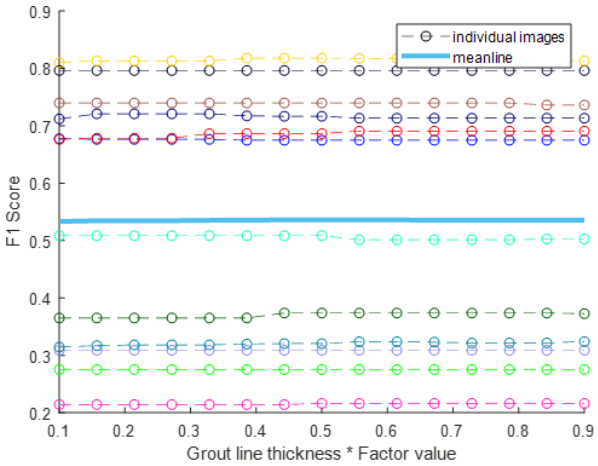
34: Incorrect mortar joint removal, vertical mortar joints confidence weighting

Vertical mortar joint detection performance:



35: Spacing error threshold for shifting singly defined vertical mortar joints

Vertical mortar joint detection performance:



APPENDIX 2: HYPERPARAMETER SEARCH FOR OPTIMISING VALIDATION ACCURACY OF DEFECT DETECTION CLASSIFIER

Input image size

batch size=50 | dense layer size=1024 | epochs for top layer training=50 | learning rate for top layer training=0.001 | epochs for fine-tuning training=50 | layers frozen during fine-tuning training=0 | learning rate for fine-tuning training=0.0001 | optimiser=SGD

top layer only:

	155	**299**
all images	0.793	0.815
brick only	0.848	0.872
categories	0.730	0.771
mortar only	0.781	0.793

fine-tuning:

	155	**299**
all images	0.867	0.866
brick only	0.903	0.900
categories	0.849	0.865
mortar only	0.850	0.849

Fully connected dense layer size

width=155 | height=155 | batch size=50 | epochs for top layer training=50 | learning rate for top layer training=0.001 | epochs for fine-tuning training=50 | layers frozen during fine-tuning training=0 | learning rate for fine-tuning training=0.0001 | optimiser=SGD

top layer only:

	2	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**	**1024**
all images	0.796	0.795	0.796	0.790	0.796	0.794	0.797	0.796	0.798	0.793
brick only	0.834	0.833	0.842	0.844	0.848	0.848	0.847	0.845	0.841	0.848
categories	0.519	0.710	0.714	0.719	0.731	0.736	0.737	0.733	0.736	0.730
mortar only	0.772	0.776	0.775	0.789	0.783	0.785	0.783	0.788	0.787	0.781

fine-tuning:

	2	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**	**1024**
all images	0.796	0.795	0.796	0.790	0.796	0.794	0.797	0.796	0.798	0.793
brick only	0.834	0.833	0.842	0.844	0.848	0.848	0.847	0.845	0.841	0.848
categories	0.519	0.710	0.714	0.719	0.731	0.736	0.737	0.733	0.736	0.730
mortar only	0.772	0.776	0.775	0.789	0.783	0.785	0.783	0.788	0.787	0.781

Top layer learning rate

width=155 | height=155 | batch size=50 | dense layer size=1024 | epochs for top layer training=50

	0.1	**0.01**	**0.001**	**0.001**	**0.0001**	**0.0001**
all images	0.500	0.500	0.793	0.791	0.792	0.785
brick only	0.500	0.500	0.848	0.848	0.853	0.851
categories	0.250	0.250	0.730	0.731	0.745	0.753
mortar only	0.500	0.500	0.781	0.776	0.778	0.768

Fine-tuning learning rate

width=155 | height=155 | batch size=50 | dense layer size=1024 | epochs for top layer training=50 | learning rate for top layer training=0.001 | epochs for fine-tuning training=50 | layers frozen during fine-tuning training=0 | optimiser=SGD

Accuracy on validation data	**top layer**	**0.01**	**0.001**	**0.0001**	**0.00001**	**0.000001**
all images	0.791	0.839	0.845	0.863	0.827	0.742
brick only	0.848	0.889	0.892	0.905	0.874	0.805
categories	0.731	0.741	0.707	0.859	0.806	0.705
mortar only	0.776	0.829	0.820	0.846	0.812	0.715

Accuracy on train data	**top layer**	**0.01**	**0.001**	**0.0001**	**0.00001**	**0.000001**
all images	0.835	0.929	0.980	0.898	0.827	0.737
brick only	0.874	0.943	0.959	0.923	0.869	0.800
categories	0.753	0.728	0.730	0.895	0.794	0.687
mortar only	0.820	0.945	0.970	0.888	0.807	0.710

Layers frozen during fine-tuning

width=155 | height=155 | batch size=50 | dense layer size=1024 | epochs for top layer training=50 | learning rate for top layer training=0.001 | epochs for fine-tuning training=50 | learning rate for fine-tuning training=0.0001 | learning rate decay rate for fine-tuning training=0 | optimiser=SGD

Accuracy on validation data:

Trainable blocks:	**0**	**all**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**
Layers frozen:	**All**	**0**	**41**	**64**	**87**	**101**	**133**	**165**	**197**	**229**	**249**
all images	0.791	0.863	0.856	0.858	0.854	0.853	0.850	0.835	0.828	0.825	0.823
brick only	0.848	0.905	0.902	0.898	0.895	0.900	0.893	0.882	0.876	0.876	0.871
categories	0.731	0.859	0.843	0.828	0.833	0.830	0.833	0.809	0.822	0.812	0.803
mortar only	0.776	0.846	0.843	0.837	0.835	0.836	0.830	0.823	0.819	0.807	0.805

Accuracy on training data:

Trainable blocks:	**0**	**all**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**
Layers frozen:	**All**	**0**	**41**	**64**	**87**	**101**	**133**	**165**	**197**	**229**	**249**
all images	0.835	0.898	0.909	0.905	0.904	0.896	0.897	0.890	0.878	0.881	0.870
brick only	0.874	0.923	0.930	0.928	0.929	0.925	0.922	0.913	0.903	0.903	0.895
categories	0.753	0.895	0.901	0.858	0.884	0.894	0.877	0.877	0.860	0.854	0.837
mortar only	0.820	0.888	0.895	0.893	0.890	0.890	0.891	0.874	0.864	0.867	0.857

Learning rate decay during fine-tune training

width=155 | height=155 | batch size=50 | dense layer size=1024 | epochs for top layer training=50 | learning rate for top layer training=0.001 | epochs for fine-tuning training=100 | layers frozen during fine-tuning training=0 | optimiser=SGD

Learning rate for fine-tuning training=0.0001

Accuracy on validation data after 100 epochs:

	top layer	**0**	**1.3E-5**	**6.6E-5**	**9.9E-5**	**1.3E-4**	**2.6E-4**	**0.12**	**0.16**	**0.2**	**0.4**	**0.8**
all images	0.791	0.853	0.856	0.861	0.861	0.859	0.852	0.681	0.680	0.670	0.648	0.646
brick only	0.848	0.900	0.903	0.903	0.896	0.892	0.882	0.738	0.730	0.714	0.709	0.681
categories	0.731	0.841	0.818	0.843	0.832	0.837	0.815	0.531	0.484	0.508	0.503	0.470
mortar only	0.776	0.834	0.846	0.847	0.849	0.844	0.837	0.644	0.640	0.636	0.618	0.612

Accuracy on training data after 100 epochs:

	top layer	**0**	**1.3E-5**	**6.6E-5**	**9.9E-5**	**1.3E-4**	**2.6E-4**	**0.12**	**0.16**	**0.2**	**0.4**	**0.8**
all images	0.835	0.956	0.925	0.886	0.876	0.869	0.856	0.676	0.667	0.667	0.650	0.637
brick only	0.874	0.954	0.939	0.917	0.910	0.902	0.893	0.739	0.718	0.709	0.703	0.672
categories	0.753	0.922	0.851	0.859	0.842	0.834	0.814	0.500	0.455	0.496	0.491	0.459
mortar only	0.820	0.942	0.914	0.872	0.864	0.856	0.842	0.641	0.644	0.636	0.622	0.611

Best accuracy on validation data and epoch number accuracy achieved at:

	top layer	**0**	**1.3E-5**	**6.6E-5**	**9.9E-5**	**1.3E-4**	**2.6E-4**	**0.12**	**0.16**	**0.2**	**0.4**	**0.8**
all images	0.791	0.866 47	0.862 70	0.864 67	0.863 77	0.860 92	0.853 90	0.683 69	0.683 87	0.671 92	0.652 97	0.649 90
brick only	0.848	0.907 72	0.907 94	0.904 99	0.897 88	0.894 77	0.884 93	0.743 97	0.732 94	0.717 96	0.714 86	0.683 94
categories	0.731	0.858 57	0.823 86	0.844 93	0.832 99	0.839 89	0.819 97	0.538 51	0.486 58	0.515 82	0.505 97	0.472 96
mortar only	0.776	0.852 45	0.851 49	0.848 95	0.850 99	0.845 94	0.837 86	0.648 65	0.645 60	0.642 67	0.625 32	0.618 47

Best accuracy on training data and epoch number accuracy achieved at:

	top layer	**0**	**1.3E-5**	**6.6E-5**	**9.9E-5**	**1.3E-4**	**2.6E-4**	**0.12**	**0.16**	**0.2**	**0.4**	**0.8**
all images	0.835	0.956 99	0.925 99	0.887 96	0.878 98	0.871 92	0.858 98	0.685 84	0.676 61	0.670 95	0.656 30	0.644 89
brick only	0.874	0.954 99	0.939 99	0.917 98	0.910 95	0.905 96	0.893 91	0.740 88	0.723 94	0.715 93	0.706 86	0.677 75
categories	0.753	0.934 95	0.851 100	0.859 100	0.843 93	0.834 100	0.816 96	0.504 90	0.457 92	0.501 8	0.492 84	0.461 74
mortar only	0.820	0.942 100	0.914 100	0.873 96	0.866 95	0.857 98	0.843 87	0.650 89	0.646 88	0.640 83	0.629 89	0.617 75

Learning rate for fine-tuning training=0.001

Accuracy on validation data after 100 epochs:

	top layer	0	1.3E-5	6.6E-5	9.9E-5	1.3E-4	2.6E-4	0.12	0.16	0.2	0.4	0.8
all images	0.791	0.861	0.853	0.853	0.842	0.856	0.848	0.770	0.765	0.757	0.723	0.707
brick only	0.848	0.884	0.891	0.895	0.879	0.898	0.899	0.817	0.809	0.800	0.786	0.757
categories	0.731	0.820	0.817	0.775	0.821	0.792	0.791	0.618	0.616	0.582	0.501	0.516
mortar only	0.776	0.824	0.829	0.835	0.831	0.832	0.844	0.752	0.753	0.736	0.704	0.684

Accuracy on training data after 100 epochs:

	top layer	0	1.3E-5	6.6E-5	9.9E-5	1.3E-4	2.6E-4	0.12	0.16	0.2	0.4	0.8
all images	0.835	0.990	0.991	0.989	0.981	0.978	0.962	0.765	0.755	0.755	0.718	0.700
brick only	0.874	0.975	0.976	0.974	0.969	0.970	0.956	0.820	0.805	0.801	0.780	0.752
categories	0.753	0.885	0.891	0.780	0.863	0.803	0.790	0.599	0.608	0.566	0.503	0.498
mortar only	0.820	0.983	0.985	0.983	0.978	0.977	0.957	0.746	0.747	0.724	0.698	0.672

Best accuracy on validation data and epoch number accuracy achieved at:

	top layer	0	1.3E-5	6.6E-5	9.9E-5	1.3E-4	2.6E-4	0.12	0.16	0.2	0.4	0.8
all images	0.791	0.863 17	0.872 10	0.866 12	0.869 13	0.873 14	0.865 18	0.772 92	0.767 98	0.759 92	0.725 94	0.707 89
brick only	0.848	0.908 47	0.908 31	0.908 28	0.902 47	0.909 36	0.901 47	0.823 93	0.812 95	0.804 99	0.787 71	0.761 98
categories	0.731	0.832 82	0.821 95	0.777 92	0.823 96	0.797 94	0.791 100	0.618 100	0.620 99	0.586 98	0.504 99	0.516 99
mortar only	0.776	0.858 12	0.853 11	0.851 12	0.852 13	0.853 11	0.854 8	0.753 98	0.757 89	0.739 83	0.705 94	0.685 91

Best accuracy on training data and epoch number accuracy achieved at:

	top layer	0	1.3E-5	6.6E-5	9.9E-5	1.3E-4	2.6E-4	0.12	0.16	0.2	0.4	0.8
all images	0.835	0.990 99	0.991 100	0.989 96	0.982 99	0.978 100	0.962 100	0.766 85	0.758 86	0.763 99	0.722 83	0.705 96
brick only	0.874	0.975 99	0.977 96	0.974 100	0.970 98	0.970 99	0.956 100	0.820 100	0.811 98	0.803 96	0.782 99	0.759 99
categories	0.753	0.885 99	0.891 100	0.780 100	0.863 96	0.804 95	0.791 97	0.601 98	0.608 100	0.570 92	0.503 100	0.502 96
mortar only	0.820	0.984 92	0.985 99	0.983 95	0.978 97	0.977 100	0.957 100	0.747 66	0.749 97	0.729 91	0.707 74	0.675 89

Fine-tuning learning rate using RMSProp optimiser

width=155 | height=155 | batch size=50 | dense layer size=1024 | epochs for top layer training=50 | learning rate for top layer training=0.001 | epochs for fine-tuning training=100 | layers frozen during fine-tuning training=0 | learning rate decay rate for fine-tuning training=0 | optimiser=RMSProp

Accuracy on validation data after 100 epochs:

	top layer	**0.00001**	**0.0001**	**0.001**	**0.01**
all images	0.791	0.848	0.854	0.856	0.828
brick only	0.848	0.885	0.887	0.885	0.879
categories	0.731	0.837	0.850	0.502	0.308
mortar only	0.776	0.833	0.814	0.832	0.810

Accuracy on training data after 100 epochs:

	top layer	**0.00001**	**0.0001**	**0.001**	**0.01**
all images	0.835	0.983	0.986	0.939	0.867
brick only	0.874	0.969	0.973	0.956	0.918
categories	0.753	0.968	0.968	0.870	0.844
mortar only	0.820	0.975	0.980	0.918	0.854

Best accuracy on validation data and epoch number accuracy achieved at:

	top layer	**0.00001**	**0.0001**	**0.001**	**0.01**
all images	0.791	0.863 18	0.867 10	0.860 40	0.852 77
brick only	0.848	0.904 33	0.908 9	0.904 69	0.885 76
categories	0.731	0.860 25	0.869 13	0.842 39	0.839 97
mortar only	0.776	0.851 23	0.856 5	0.849 55	0.836 80

Best accuracy on training data and epoch number accuracy achieved at:

	top layer	**0.00001**	**0.0001**	**0.001**	**0.01**
all images	0.835	0.984 99	0.987 99	0.939 100	0.867 100
brick only	0.874	0.970 99	0.973 87	0.957 97	0.918 100
categories	0.753	0.968 100	0.968 92	0.884 99	0.851 97
mortar only	0.820	0.975 100	0.980 97	0.919 98	0.854 100