UNIVERSITY OF SOUTHERN QUEENSLAND

Faculty of Health, Engineering and Sciences

A REVIEW AND ANALYSIS OF NEUROMORPHIC COMPUTING TECHNOLOGY TO DETECT CONCRETE STRUCTURE DEFECTS USING OBJECT DETECTION

A dissertation progress report submitted by

Allan Bourke

in fulfilment of the requirements of:

ENG4111/2 Research Project

Towards the degree of

Bachelor of Engineering Honours (Civil)

Date submitted: 15 October 2023

Abstract

Deep learning utilises neural network layers to systematically process similarities and differences between images to establish a reliable set of features used to assist in localising and classifying objects within an image. This is done through an object detection model and the output is an application of a class label to the object within the image. Object detection models could automate, and therefore greatly expedite, the inspection of concrete structures which are currently periodically visually inspected by trained inspectors to determine asset condition and maintenance and requirements. This manual process is considered 'Gold Standard'; however it is time-consuming, expensive, and triggers field hazards. To date, many research studies have demonstrated successful use of second-generation artificial intelligence (AI) technology to identify defects in concrete structures, attempting to replicate the inspector. More recently, advances in AI systems have led to the development of third-generation neuromorphic computing technology and its use for object detection has the potential to increase detection speed and efficiency, use less power, enhance security, and allow for data to be analysed locally without requiring large cloud-based data stores. The overarching aim of this project was to demonstrate that neuromorphic computing technology is a suitable novel technology to detect common defects on concrete bridge and culvert structures using object detection. Specifically, the project aimed to develop, train and implement a neuromorphic computer vision model to identify common bridge defects and determine the system accuracy, effectiveness and usability. The model was also directly compared with a traditional object detection model.

Photographic images (n = 844) of concrete structures were manually collected through field inspections and dissected into 17395 512 x 512 pixel images. These were manually classified to obtain 1326 images of various structure defects. Of these, 200 high quality crack and spall images were utilised to develop YOLOv5 and AKIDA object detection models using Edge Impulse studio with three hundred (300) training cycles. The model configuration was set using a bounding box labelling method. For all models, 64% of images were allocated to training, 16% for validation, and 20% for model testing. Crack only models were developed using 100, 150, 170, 200, 350, 300 and 350 images and a spall and crack model using 100 spall and 100 crack images was also developed. The learning rate varied between 0.01 and 0.001 for AKIDA to optimise testing results while the learning rate of 0.01 for the YOLO5 model could not be modified through the online portal. Confidence thresholds of 30%, 50% and 70%t were set to analyse the accuracy of these models using the test dataset. Finally, the performance of both models was visually assessed in real-time environment replicating a field setting.

Both models produced poor results overall for the combined crack and spall image dataset, with less than 30% accuracy using the lowest confidence threshold. Further, they produced 0% accuracy for spalls using all three (3) confidence thresholds specified. However, both models were able to successfully detect concrete cracks in the images. The AKIDA model produced the highest precision

(94.1%) of all models on the validation data, using 150 images, while YOLO only achieved 69%. The highest precision able to be obtained with the YOLOv5 model was 75% using 350 crack images. For accuracy, the AKIDA model achieved 81.7% accuracy with 300 images using the 30% confidence threshold and 68% accuracy under the 70% threshold. The highest accuracy achieved by YOLOv5 was 25% using 200 images. Conversely, the YOLO model performed better in the visual inspection test, with more accurate crack detection in 10 out of 14 instances through this subject assessment process.

The study highlighted that image quality can negatively affect results and model development. In particular, neither model was able to successfully detect spalls, as these are a more complex defect, making them more difficult for the models to detect. However, both models were able to detect cracks, with the AKIDA model demonstrating that third-generation models were comparable to second-generation technologies for the identification of cracks in this setting. Further investigation could incorporate additional defects into the object detection model, to simulate a comprehensive structure inspection. The study suggests neuromorphic computing is a new promising technology to identify common defects in concrete structures.

Disclaimer

University of Southern Queensland

Faculty of Health, Engineering and Sciences

ENG4111 & ENG4112 Research Project

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Health, Engineering and Sciences, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Health, Engineering and Sciences or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitles "Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and any other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Candidate Certification

I certify that the ideas, designs and experimental work, results, analysis and conclusions set out in this dissertation are entirely my own efforts, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Allan Bourke

Student Number: [redacted]

[redacted] **(Signature)**

15 October 2023 **(Date)**

Acknowledgements

I would like to thank Drs. Andy Nguyen and Jason Brown from the University of Southern Queensland for allowing me to research this dissertation under their supervision.

As those special to me know, getting to this point has been a life long journey. To mum, late dad and nan, this paper is for you all. Also, my beautiful wife Mel, my daughters Jasmine and Sheridan, and our dog Bella, thank you for all your ongoing support. The joint frustration you must share when assignment deadlines approach and you realise that I've only just downloaded the brief.

Table of contents

List of tables

List of figures

List of appendices

Chapter 1: Introduction

1.1 Background

Australia's arterial road and bridge maintenance expenditure for 2019-20 was estimated to be $2.8 billion. Expenditure for Queensland alone was estimated at $725 million (Bureau of Infrastructure Transport and Regional Economics (BITRE) 2021). As part of this maintenance program bridges and culverts located on Queensland state-controlled roads are inspected periodically in accordance with the Department of Transport and Main Roads (TMR) Structures Inspection Manual. Inspection data are utilised by TMR to determine asset condition and assist in planning for future asset maintenance and rehabilitation (State of Queensland (Department of Transport and Main Roads) 2016a).

Traditional methods for inspection entail the use of trained inspectors to carefully examine structures at arm's length; photographing, measuring, and documenting defects and asset condition (Mirzazade et al. 2023). Inspectors must also consider the need for traffic control, management of traffic disruptions, safety within the road corridor, and methods for inspecting hard-to-access locations (i.e., at height, confined space, or below water) (Flah, Suleiman & Nehdi 2020; Mirzazade et al. 2023). These factors make visual inspection techniques time-consuming and costly for asset holders.

In recent years, there has been a substantial increase in the use of artificial intelligence (AI) technologies in the field of structural engineering (Hadi & Rigoberto 2018; Flah, Suleiman & Nehdi 2020; Huu-Tai 2022). With developing technologies, there is potential for AI, and in particular the AI technique of deep learning, to compliment traditional structure inspection techniques, and to aid in rapid decision making at a reduced cost (Hadi & Rigoberto 2018; Flah, Suleiman & Nehdi 2020). Combined with the use of Remote Piloted Aircraft (RPA), there is potential to fully automate the inspection process and eliminate the need for the existing 'Gold Standard' manual inspection method (Guido et al. 2019; Flah, Suleiman & Nehdi 2020).

"AI leverages computers and machines to mimic the problem-solving and decision-making capabilities of the human mind" according to the pioneers of advanced computing; International Business Machines Corporation (IBM Cloud Education 2020a). The key parameter that makes the current AI technologies particularly applicable to engineering applications is that they are underpinned by machine learning (Jordan & Mitchell 2015; Huu-Tai 2022). Machine learning is a sub-discipline of AI where computer systems are exposed to extensive labelled datasets that allow them to identify successful patterns, with each iteration progressively refined until endpoint decision making is accurate and repeatable, without the need for explicit programming (Jordan & Mitchell 2015; Huu-Tai 2022).

Several statistical methods can be used by machine learning, including linear or logistic regression, neural networks and decision trees (Alqahtani & Whyte 2013; Jordan & Mitchell 2015). Most often machine learning makes use of artificial neural networks that essentially mimic the biological processing function of the brain, by using a set of algorithms (Alqahtani & Whyte 2013; Hadi & Rigoberto 2018; IBM Cloud Education 2020a). A network consists of input data, weights that affect the strength of the signals sent and received, a bias, and an output. An algebraic formula is used to determine whether the output is greater than a specified threshold. If the threshold is met or exceeded then the node forming that part of the network is activated, and the data are then passed to the next layer (IBM Cloud Education 2020a). The more network layers, the deeper the network. Once all network hidden layers are determined, they are passed to a final layer to calculate the final network layer output and form a decision (IBM Cloud Education 2020b). Currently, the precision and accuracy of these AI-generated outcomes is variable, depending on the size and complexity of the training dataset, and the difficulty of the task. An area of rapid growth is the use of deep learning for the automated detection of objects in images, and this was the specific focus of this research project.

1.2 Application of Research

As a result of this project, it is expected that the hypothesis will be proven and the project will demonstrate similar accuracies and precision between a neuromorphic object detection model and a 2nd generation counterpart. Given the ability for neuromorphic models to be run using less power and less latency, the neuromorphic alternative is predicted to be better suited to implementation with remote piloted aircraft (RPA's) with future potential to fully automate the inspection process. Automating the inspection process could lead to faster data acquisition and processing, enabling more timely decision making for asset inspectors and owners. Added benefits of automated inspection include improved safety by reducing human interactions with traffic, fewer disruptions to the traffic network, and the ability via the use of RPA's to access hard to reach locations.

1.3 Aims and objectives

The principal aim of this project was to demonstrate that neuromorphic computing technology is a suitable novel technology to detect common defects on concrete bridge and culvert structures using object detection.

The specific objectives were to:

- Review current concrete bridge and culvert defect inspection practices
- Review neuromorphic computing technology use for object detection
- Establish a methodology to incorporate neuromorphic computing through deep learning to identify concrete bridge and culvert defects

- Inspect and photograph common bridge and culvert defects for use in the object detection system development
- Develop and train a neuromorphic computer vision model to identify common bridge and culvert defects
- Implement the object detection model through a field trial
- Analyse the field trial results to determine the system accuracy, effectiveness and usability
- Consider improvements to increase the system accuracy, effectiveness, and usability
- Compare the neuromorphic model with a traditional object detection model

1.4 Scope

This project will deliver a customised object detection model to detect two or more common concrete bridge and culvert defect types using neuromorphic computing technology. The project will also evaluate the model against a comparable object detection model trained using the same image dataset. The scope of the project is intentionally narrow, to ensure the accuracy and precision of the selected technology. This approach will increase the likelihood that the outcomes can be further developed and adapted to a more diverse range of defects affecting concrete structures.

Chapter 2: Literature review

2.1 Common defects of concrete structures

Concrete, along with steel, is one the of the most common construction materials used worldwide mainly due to its impressive ability to resist compression (Zhang et al. 2018; Yin et al. 2021). In the transport engineering sector, concrete is the key material used in the construction of many structures, such as bridges and culverts. However, concrete is also prone to structural defects that can occur due to aging and/or damage (Roper, Kirkby & Baweja 1986; Zhang et al. 2018). This project focuses on common defects of concrete structures, such as cracks and spalls. Defects in concrete structures manifest from a variety of stressors placed on the structure, such as temperature extremes, natural disasters, high impact collisions and excessive loading, and are likely to increase in frequency with the age of the structure (Roper, Kirkby & Baweja 1986; Yin et al. 2021). Common deterioration mechanisms are outlined in the TMR Structures Inspection Manual Part 2 (State of Queensland (Department of Transport and Main Roads) 2016b). These mechanisms can affect most parts of the concrete structure and can include a) Corrosion of the reinforcement material; b) Alkali-Aggregate Reaction (AAR); c) Cracking; d) Spalling; e) Delamination; f) Surface Defects; g) Scaling; h) Disintegration; and i) Fire damage (State of Queensland (Department of Transport and Main Roads) 2016b). Other common defects identified during the authors field inspections include graffiti from an aesthetics perspective and damage or deterioration to the pavement wearing surface. Photographs showing some defect images photographed in the field are provided in Figure 1 below.

Figure 1: Various defect types identified during field inspections
Images courtesy of: A. Bourke

Cracking and spalling are two of the most common concrete structure defects identified during routine inspections (Roper, Kirkby & Baweja 1986). A crack is defined as a separation that occurs in the surface of the concrete, dividing the structure into one or more segments (Yin et al. 2021). Cracks can be structural (e.g., resulting from loading or impact) or non-structural (e.g., resulting from temperature or incorrect curing) in nature (Yin et al. 2021). Concrete spalls occur when the surface of the concrete corrodes and fragments, and this type of defect can result from fatigue of the concrete surface (Yin et al. 2021). This project will focus on the detection of cracks and spalls and the reader is referred to the descriptions provided in the TMR inspection manual (State of Queensland (Department of Transport and Main Roads) 2016b) for further information about the other defect types.

2.2 Assessment methodologies for concrete structure defects

2.2.1 Visual Structure Inspections

Accurate and regular assessment of concrete structures to identify defects early in their development is very important for timely intervention and remediation (Roper, Kirkby & Baweja 1986). The most common traditional technique for the assessment of structure condition involves visual inspection by trained inspectors. This visual inspection methodology is very low impact on the structure and has been considered the 'Gold Standard' for concrete structure inspection worldwide, but this approach is time-consuming, expensive, and sometimes dangerous (Flah, Suleiman & Nehdi 2020; Mirzazade et al. 2023).

In Queensland, Australia, the condition of structures within the TMR road corridor are assessed and monitored through periodic visual inspections. The inspection process is regulated, with inspectors required to undergo training, and follow detailed protocols (State of Queensland (Department of Transport and Main Roads) 2016a, 2016b, 2016c). There are three (3) levels of inspection applied to road structures, with their frequency and description detailed in the table below:

Table 1: TMR Structure Inspection Types

Type	Typical Frequency	Description
Level 1	Annually	The level 1 inspection is classified as a routine maintenance inspection. It is typically a visual inspection to check for general wear and tear, and to look for safety or emerging issues.
Level 2	Every 2 to 5 years	The level 2 inspection involves a condition rating of the structure. It is used to assess previous maintenance performance, highlight current maintenance issues, and used for future condition modelling and budget forecasting.
Level 3	When determined by a Level 2 inspection	The level 3 inspection is used to address other issues beyond the scope of the level 1 and 2 inspections. The categories include: • Structure engineering

		• Asbestos containing material (ACM) identification • ACM verification • Underwater access • Fracture critical/low redundancy (relevant to steel bridges only) • Sub-standard load rating • Complex/unique structures • Known/suspected deficiencies • Confined space inspection

Source: (State of Queensland (Department of Transport and Main Roads) 2016c)

Procedures to carry out TMR structure inspections are contained within Part 3 of the Manual (State of Queensland (Department of Transport and Main Roads) 2016c). The appendices of this manual contain an inspection report template that is used by inspectors completing their visual inspections in addition to the Structure Standard Component Identification Schedule which ensures uniformity of reporting. Adherence to these procedures is essential to minimise variability of reporting amongst inspectors, which, as with any manual technique, is subject to bias.

There are numerous factors that can influence the quality of visual inspection. Stallard et al (2018) in their study of visual error for metal casting inspections state that the error factors included training, type of judgement used, percentage of defects, environmental conditions, and the inspector's ability (Stallard, Cameron & Frank 2018). Using their probabilistic model, they found error rates of up to 40% for false alarms, and 35% for missing the defect. This reasonably high potential for inter-inspector variability and bias undermines the accuracy and precision of visual inspection techniques.

2.2.2 Manual inspections

Manual inspection techniques include methods to assess the severity of a defect that is detected during a visual inspection, and thus most commonly occurs in conjunction with, or soon after, a visual inspection. Traditional methodologies are quantitative and involve measuring and documenting of the severity and progression of defects. The necessity for a manual inspection would typically be determined during a visual inspection, and the size of a defect is a key parameter used to determine the frequency of subsequent assessments and/or the need for remediation. Accurate documentation of the extent of the defect enables progressive condition assessment of a structure, so that interventions to remediate the defect can be planned for and occur in a timely way. These physical assessments still require trained inspectors to record the data and remain a key component of structural health monitoring programs for infrastructure. Methods range from simple, such a physical measurement of the length and width of a defect, to more advanced techniques, such as computer vision methods that automatically capture and record defects (see section 2.2.4).

The key benefit to a manual inspection is the ability to grade a defect, so that an ongoing assessment of deterioration is less subjective. The grading scales used are usually simple, such as a numerical scale of 1 - 4, and are mandated by the responsible authority (Christian et al. 2015). Measurements, such as the length of a crack, enable the defect to be graded, and cumulative scores result in an overall grade being assigned, such as good, fair, poor or severe (Christian et al. 2015). The progression of the condition of a structure can then be monitored with respect to the speed and degree of deterioration. While manual inspection reduces the degree of bias compared to visual inspection, more advanced quantitative techniques, such a non-destructive technologies and mathematical modelling, are being used with greater frequency (Flah, Suleiman & Nehdi 2020).

2.2.3 Non-destructive technologies

To extend on manual measurement methods for the assessment of civil structures, many non-destructive methods have been developed. These methods enable assessments to go beyond simply assessing the surface of the structure and include thermography, ultrasound, tomography, radiography, electromagnetic and electrochemical methods (Balayssac & Garnier 2017). The techniques can be applied individually, but are more often used in combination, and the choice of test depends on the substrate material and type and extent of the defect (Balayssac & Garnier 2017). As an example, ultrasound is a technique that has been used to assess the condition of solid concrete structures, as the sound waves that pass through the dense material can give an indication of the resistance i.e., strength, of the internal environment of the solid structure (Hu et al. 2021). The parallels to human medicine are obvious, with similar methods used to examine the health of the less accessible internal parts of the human body.

The output data obtained from non-destructive testing can be collated into combined datasets and mathematical models and complex algorithms can be used to determine the significance of the data (Hu et al. 2021). Increasingly, AI methodologies, such as neural networks, are being employed to analyse data derived from non-destructive techniques (Schabowicz 2019; Hu et al. 2021). One major advantage of using non-destructive methods is that it enables repeat testing, so outputs can be used to assess structures over time and cumulative datasets can be analysed with powerful statistical models (Schabowicz 2019). A disadvantage of non-destructive testing methods is that they are heavily influenced by weather, e.g., heat and moisture, so conditions for testing must be comparable, which affects the planning and execution of testing. Non-destructive technologies can also be expensive as they rely on specialty equipment, and require highly trained personnel, reducing their feasibility for some operators (Schabowicz 2019). Further, the need for extensive data processing can increase the time taken to reach decisions about the health of the structure. Semi-destructive, such as core sampling, and destructive techniques are also used for the assessment of concrete structures but are beyond the scope of the current study.

2.2.4 Computer vision methods

Visual structure inspections produce large banks of digital images in order to record the number, type and extent of concrete defects (Ren et al. 2022). A need for improved methods for processing and cataloguing these images has resulted in increasing use of computer vision technologies for civil infrastructure inspections (Christian et al. 2015; Ren et al. 2022). In recent years, use of computer vision methodologies has enabled automation of the process where images can be processed without the need for human input. The ultimate goal of applying computer vision methods to a civil structure inspection process is to enable 1) the detection of an object, e.g., a concrete crack, in an image and 2) the analysis of the nature of the object. The images can be acquired during visual inspections, or could be captured remotely, e.g., by a remote piloted aircraft. The degree of image processing varies, with low level processing aimed at ensuring the image quality is sufficient for subsequent analysis and intermediate-to-high level processing focused on identifying and extracting features of interest to enable analysis (Christian et al. 2015; Ren et al. 2022). Computer vision methods can be feature-, model- or pattern-based (i.e., non-neural approaches), or rely on a 3D reconstruction of the defect (neural approaches) (Christian et al. 2015). With respect to the identification and assessment of concrete cracks, many of these methods have been investigated in studies aiming to assess their applicability to automated crack detection in large concretes structures, such as bridges (Zakaria, Karaaslan & Catbas 2022; Zhang, Karim & Qin 2022).

Feature-based methods, such as the Haar-wavelets technique, use edge detection algorithms to identify cracks (Hoang, Quoc-Lam & Van-Duc 2018; Olisa et al. 2018). By comparison, model-based methods focus on segmentation, where the image is separated into defect and non-defect areas, while pattern-based methods rely on object detection and principal components analysis (Kim et al. 2022). These methods can be used alone or in combination, and many systems have been developed and investigated for crack detection and assessment, with varying levels of success (Olisa et al. 2018; Zakaria, Karaaslan & Catbas 2022; Zhang, Karim & Qin 2022). All of these methods are subject to error, based on the quality of the image taken, the suitability of the image collection used for training the model and the power of the machine learning model (Christian et al. 2015; Ren et al. 2022). Variations in image quality i.e., clarity, camera angle, distance from the object, exposure, contrast and saturation will all significantly impact the output data (Ren et al. 2022). Further, capturing images in exactly the same way each time a structure is re-assessed is nearly impossible. As such, it is important that the methods used for image analysis are able to correct for variations in image quality and acquisition (Ren et al. 2022). This is potentially best achieved if the image analysis method can be 'taught' to account for these parameters, which is achieved by training the model with large quantities of images (Kumar et al. 2021; Kim et al. 2022; Zhang, Karim & Qin 2022). Machine learning methods using neural networks are 3D reconstruction-based methods that can achieve this 'intelligent' image processing (Dung & Le Duc 2018; Kumar et al. 2021) and are the computer vision technology used in this project. Obviously, it is

also crucial that images can be accurately traced back to their location, and the most common approach for achieving this is the use of satellite navigation and the global positioning system (Miles 2023), which enables placement of exact coordinates on each image. This is an area of active research with substantial refinement still required to ensure accurate, repeatable and precise outcomes.

2.3 Artificial intelligence and civil infrastructure

This study focuses on identifying defects in concrete structures using AI deep learning and object detection. In particular, this study will determine if third-generation AI is as effective as its second-generation counterpart in identifying concrete structure defects. However, prior to reviewing and synthesizing the complex methodologies required, a review of the development and history of AI, with respect to civil infrastructure inspection, is provided.

2.3.1 The generational development of artificial intelligence

It is well known that computers are replacing their human counterparts for many and varied tasks. The AI technologies arise from a sophisticated fusion of neuroscience and computer science. As such, opportunities for the application of AI are considerable in civil engineering, where complex problems often require a combination of human initiative and mathematical modelling (Hooda et al. 2021). There have been numerous attempts to utilise AI technology to identify defects in concrete structures.

The original generation of AI technologies did not have the advantage of utilising a statistical approach, relying on a deep, but static, underpinning knowledge. The systems needed to rely on these large databases in order to solve problems, and as such were more structured and did not enable a flexible approach to novel problems. This minimised the breadth of potential problems that the first-generation systems were able to solve. Problems that were appropriate for this early form of AI were optimizations of billing and supply and demand systems, and the approach was used to reduce overheads and predict financial positions going forward. In this situation they were still an intelligent system but lacked the ability to take an abstract view of repeated problems.

The second-generation of AI systems is still the most heavily utilised and has improved on the first-generation technology by introducing statistical learning to broaden the scope of problems that can be solved, and starting to increase the ability to solve novel applications without reliance on expert input (Bilgil & Altun 2008). Most readers would be familiar with second-generation technologies as they underscore machine learning, and power well-known applications such as Siri and Alexa. The introduction of statistical approaches to intelligent systems has revolutionised the scope of their use, which stretches from health to music to education (Miguel et al. 2022). They are even used to stretch players of games of strategy, such as Chess. There is still room for improvement though, given that a

computer recently broke the finger of a 7-year-old boy that it incorrectly deduced to be cheating (Angelova & McCluskey 2022).

Many research studies have demonstrated and investigated the successful use of second-generation AI technology for computer vision tasks, such as concrete defect detection as outlined above (Dung & Le Duc 2018; Kumar et al. 2021). They make use of large, cloud-based data centres, but do have a requirement for huge power consumption and their speed is limited by the power and capability of the processing system being used. Use of second-generation technology is still a growing area of engineering science, and perhaps remains the focus for most research despite the recent progression to third-generation AI technology.

Building on the success of second-generation technologies requires the implementation of complex thinking, and the ability of the system to not only solve a task, but to do so in a novel way that has incorporated a critical assessment of the approach to the problem. The final step is then to translate the approach used to the human operator, so that they can understand the approach taken and the rationale for its use. Accordingly, third-generation systems are still largely developmental and experimental in nature, although their development has made huge leaps in recent years, with many more examples of their successful use (Hooda et al. 2021). One such example, as outlined in the introduction is the use of neuromorphic computing by companies such as Intel, BrainChip and Nvidia. These companies seek to use third-generation AI technology to mimic the human brain to solve complex, but everyday problems, including disease detection and speech recognition and incorporate these skills into larger multi-faceted approaches to layered problems that require a sequential, cognitive approach to solve successfully. One application relevant to this project is object detection and classification (Christian et al. 2015). This application has been well researched for second-generation approaches, and while research persists the ability to include a cognitive element to solving the problem of large-scale object detection has instigated research into possible uses of third-generation AI techniques in this area of engineering science (Hoang, Quoc-Lam & Van-Duc 2018; Kim et al. 2022).

A final key advantage of the third-generation of AI is its suitability for edge based technologies, otherwise known as edge computing. Essentially, the data can be analysed locally, without the need to access a large, centralised data processing unit, enabling better reactivity to data outputs e.g., smart watches. This greatly increases speed and efficiency when it comes to neuromorphic computing applications. Gartner states that in 2018 approximately 90% of all enterprise-generated data was managed in centralized data centres or the cloud (Gartner 2018). By 2025 it is anticipated that edge base technologies will manage up to 75% of the enterprise data (Yin et al. 2021). There is still a place for second-generation large cloud datacentres, using powerful and power intensive computers to run AI models. BrainChip state that the generational technologies will continue to work together, forming distributed optimised models fit for purpose (Brainchip 2022a).

2.3.2 Neuromorphic computing

More recent development of AI technologies has extended machine learning via neural networks with the use of larger, unlabelled datasets, in a sub-discipline called deep learning (IBM Cloud Education 2020c). The critical factor that differentiates deep learning is its capacity to use unstructured data, e.g., images. Using at least three neural network layers the data are systematically processed until a reliable set of features are determined that enable the data to be categorised (such as recognising the similarities and differences between images), bypassing the need for human input (Hooda et al. 2021). Intel state that current second-generation AI technologies deal with perception and sensing but can also use deep learning networks for image data recognition (Intel 2023). However, more current terminology refers to deep learning technologies as neuromorphic computing, or third-generation AI which goes beyond the simple network patterns that constitute second-generation AI and attempts to replicate the human brain on a deeper level, to deal with uncertainty, ambiguity, and contradiction (Hooda et al. 2021). As the field evolves, these advances will assist in overcoming the current shortfalls with AI and deep learning that essentially lack a critical assessment of the task (Intel 2023).

Neuromorphic computers are defined as non-von Neumann computers which operate like the human brain in structure and function. Processing and memory are controlled by neurons and synapses, with programs governed by the neural network structure; whereas Von Neumann computers consist of separate processing and memory units, with programs executed by a series of instructions. Neuromorphic computers use the magnitude, shape and time of occurrence of spiking neurons to encode data, whereas von Neumann computers encode numerical values in binary form (Schuman et al. 2022).

The Neuromorphic Computing Market was estimated to be worth $34.23 billion in 2022, increasing to $225 billion by 2027, with an annual growth rate of 45.8% (Markets and Markets 2021). It is suggested that image recognition will be responsible for the largest portion of the market during this time period, mainly due to the increase in use of RPA's, video monitoring and computer vision (Markets and Markets 2021). One of the leading neuromorphic market participants, BrainChip Holdings Ltd. (US), highlight that a key benefit of neuromorphic computing over second-generation AI is the potential for incremental learning that relies on fewer datasets to reach a decision, with reduced time to make that decision (Brainchip 2022a). Additional benefits include improved efficiency (up to a 1000-fold reduction on power consumption) and security (Brainchip 2022a). The leading neuromorphic market participants are said to be Intel Corp. (US), IBM Corporation (US), BrainChip Holdings Ltd. (US), Qualcomm (US) and HP Enterprise (US). Clearly these advances have the potential to revolutionise the process of identifying defects in civil structures, by using neuromorphic computing to quickly and accurately detect defects in a series of images from a structure using object detection.

2.3.3 Object detection

The typical deep learning (neural network based) object detection process referred to above involves image localisation followed by image classification. Image localisation is the process whereby machine learning is utilised to establish where objects are located in an image. Generally, objects are defined by a set of bounding box coordinates, or single coordinate and bounding dimensions (Syed Sahil Abbas et al. 2022). Image classification is the next stage in the progress, whereby machine learning determines what type of object is within the bounding box area. The output following image classification is the application of a class label to the image (Xiongwei, Doyen & Steven 2020). The combination of these two processes comprises object detection, where the object in question is located and classified by an algorithm, with typical output shown in the figure below.

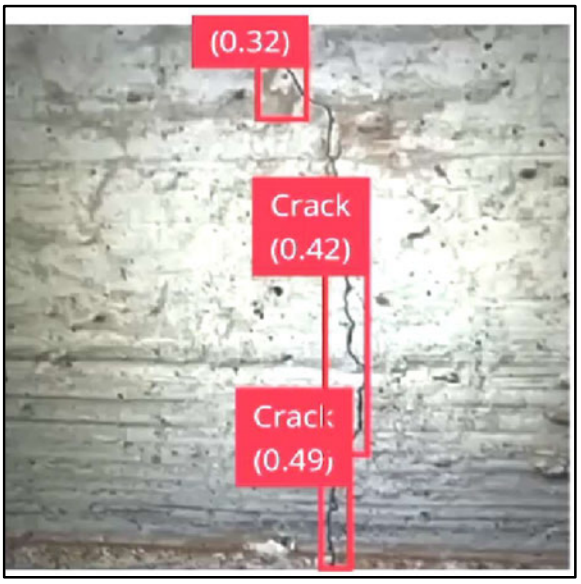


Figure 2: Object detection output - localisation and classification
Screenshot courtesy of: A. Bourke, content of image sourced from model developed at: (*Edge Impulse - Optimize AI for the edge* 2023)

The key advantage of current object detection models is that they do not require specialised camera equipment and can process images captured on any digital camera. This has helped to make the technologies more accessible for users without access to large scale computing power, a significant disadvantage of early object detection methods. However, the main disadvantage of deep-learning based object detection remains the extensive datasets required to train the models. Several open access datasets have been published, such as Open Images provided by Google®. However, these large, well-developed datasets are not focussed on object detection with respect to civil structures so most engineering researchers have developed their own image training datasets, specifically focussed on their requirements of their studies.

There are several neural approaches available for object detection, most of which are based on deep convolutional neural networks. The earlier, two-step models such as the region convolutional neural network (RCNN), were the pioneering models that provided major advances in the field of object detection, but have largely been superseded by the one-step models that rely on aspect ratios, including You Only Look Once (YOLO) and Retina-Net (Syed Sahil Abbas et al. 2022). These one-step techniques increase the speed and efficiency of object recognition for real-world applications because they do not process that data in two separate steps, but do carry the risk of reduced accuracy (Huang et al. 2017). However, due to the numerous improvements made to both (including Faster-RCNN and YOLOv3) both technologies still hold a place in varied applications (Yang et al. 2021).

Another recent development is the use of the Faster Objects More Objects (FOMO) based algorithm for object detection. This algorithm differs from traditional architecture in its final network layer by producing a heatmap of object locations based on a probability map for each region of the image and a loss function preserving locality (Dickson 2022). Output results are the object classification and centroid, as opposed to classification and bounding boxes. It is estimated that a FOMO neural network operating on a Raspberry Pi 4 can detect objects up to 30 times faster than MobileNet SSD (Dickson 2022).

Regardless, accuracy remains an important limitation for all object detection methods, given that an 'object', such as a concrete crack, can take many different forms, angles, and dimensions.

2.3.4 Deep learning models used in this project

2.3.4.1 YOLOv5

The second-generation deep learning model to be used in this project is YOLOv5, a development from YOLO (Redmon 2016) which had traded accuracy (to some extent) for superior speed of object detection and therefore suitability to real-time detection scenarios, such as the field-based applications investigated in this project (Xiongwei, Doyen & Steven 2020). YOLOv5 has been selected because it (and its' predecessors) has been widely tested for many applications and has been used successfully for the identification and localisation of defects in civil infrastructure previously (Kumar et al. 2021; Xiaoning et al. 2021; Chen, L. et al. 2023). This model will take the role of the positive control object detection technology, that the selected neuromorphic computing model, AKIDA (Brainchip 2023b), will be compared against.

As a result of its superior speed and accuracy compared with other models, YOLO, a single stage detector, is used commonly in engineering applications. For example, in a recent study YOLOv3 was reported to have better than 95% accuracy and precision for the detection of erosion in a concrete structure (Xiaoning et al. 2021). In addition, YOLO has been compared with non-destructive testing

methods, in this case projected laser, with favourable outcomes with respect to the speed and efficient of detection of concrete cracks (Song Ee, Seung-Hyun & Haemin 2020).

The YOLO model divides an image into cells in a grid pattern and uses regression methods to pinpoint the centre of the object. The key difference with this deep learning model is that it detects the object at the level of pixels, rather than relying on the bounding box co-ordinates (Xiongwei, Doyen & Steven 2020). Limitations relating to the ability to detect multiple objects co-located within a cell are the key issue for users of YOLO, although these issues have been addressed in updated versions of the program (Syed Sahil Abbas et al. 2022). Typically, because images of defects in civil structures, such as concrete cracks, are often the only object of interest in the image, YOLO can perform well with structure defect datasets based on cracks and spalls.

Several iterations of the program have been released since the original version, each with improvements over its predecessor with the latest versions including edge detection technology (the identification of the boundaries, or edges, between objects in an image). These later versions (up to YOLOv3) include significantly optimised additional features, such as batch normalisation and the use of anchor boxes (which enhances recall), aimed at improving the capacity of YOLO to decipher more complex or crowded images (Syed Sahil Abbas et al. 2022). However, the detection of small objects in particular remains a significant challenge (Yang et al. 2021). The refinement of YOLO continues with each iteration faster and more accurate than previous version (currently up to YOLOv8), which continues to see YOLO a strong competitor in the deep learning model space.

The YOLOv5 model for this project is developed through the Edge Impulse online machine learning development portal, using specifically the YOLOv5 nano community architecture. The model architecture is pretrained on the COCO dataset (GitHub Inc. 2023) and developed through Ultralytics research (Jocher 2020).

2.3.4.2 AKIDA

AKIDA (Brainchip 2023a) is a digital neuromorphic processor made by BrainChip® that is hardware-based. This neural network model enhances the speed and efficiency of information processing by using an on-chip system that leverages spiking neural networks (SNN). The advanced design technology of AKIDA addresses two of the key drawbacks common to most of the systems used in engineering applications to date: the slow speed of processing, and storing, data and the large amounts of power required to drive processing. There are several other chip-based neuromorphic programs that could be used for engineering applications, such as TrueNorth (IBM), DYNAP and Loihi (Intel), and the competitive nature of this research field is driving innovation and rapid advancement. There were two main drivers for selecting AKIDA for use in the current project: it is commercially available and was developed in Australia (Brainchip 2023a).

Early applications of the AKIDA technology were in the sensory data fields where it was used to process (and predict) vision, olfactory and auditory outcomes (Vanarse et al. 2022). The use of AKIDA in the field of civil infrastructure is in the very early stages, and the focus is on visual applications, aiming to use AKIDA for problems suited to computer vision solutions. One recent report stated that the accuracy of the AKIDA system was 82% when it was asked to perform a classification test using the author's dataset, which differs considerably from the 97% accuracy reported by BrainChip (Park & Kim 2021; Brainchip 2023a). Such differences likely derive from the quality of the datasets used to train the system. The true value of these systems though is their capacity to learn continually and improve accuracy over time (Brainchip 2023a).

The SNN approach mimics the way that neurons in the brain fire, where each neuron communicates with the next using an electrical charge generated by the flow of charged ions, called an action potential. Actions potentials move along nerves asynchronously, and only fire when a threshold is reached indicating that information needs to be processed (Marieb & Hoehn 2019). In the AKIDA model the "neurons" are arranged in a parallel design and mimic this impulse based process using a voltage change that exceeds a threshold charge to trigger data spikes of processing. This design greatly reduces the power required to perform the processing, and this is the key improvement that models using SNNs bring (Ivanov et al. 2022). However, given that the average processing exercise costs the human brain only 10-20 watts of power, and most of the brain is quiescent at any given time, neuromorphic computing still has room for improvement (Kováč 2010). Further, a key limitation with the use of AKIDA, and all systems that rely on SNN, is that it can be significantly derailed by any interruption during the spikes of communication, which reduces accuracy (Ivanov et al. 2022).

By being chip-based (edge AI) AKIDA aims to overcome the limitation of high processing power requirements of software-based neural systems. When the processing and memory elements of a system are physically separated, the interchange of data between the 'off-site' memory and the processor is the rate-limiting factor for the system. Having two separate systems (or locations) required to drive the computing process also increases the power requirement (Horowitz 2014). By using a local learning system, AKIDA can engage continuous learning directly on the chip which increases speed and reduces power costs. The on-chip learning not only speeds up processing but enables more agility and responsiveness and can also enhance security, as there is no requirement for data to be stored in the cloud (Ivanov et al. 2022).

Neuromorphic systems are heavily reliant on algorithms and are affected by the fact that the periphery of the output models will always be more variable; therefore, predictions based on outlying data will be less accurate than if the input data sits in the core of the mathematical model. At present the training of any SNN-based model is complicated by the spike-based nature of neuron firing (Siddique, Vai & Pun 2023). How do you train the neuron when to fire? Development of SNN specific algorithms for training

of chip-based models is an area of rapid development and is crucial to improving the accuracy of these systems (Siddique, Vai & Pun 2023). Customised training algorithms will also help to address the issue of 'noisy' datasets which is an important area of refinement needed for more repeatable object detection in images.

The AKIDA model for this project is also developed through Edge Impulse (*Edge Impulse - Optimize AI for the edge* 2023), using an AKIDA FOMO architecture. The model architecture was limited by the available models through Edge Impulse that could be deployed to an AKIDA system on chip PCIe card which can be deployed to a Raspberry Pi development kit. The model is pretrained using an AkidaNet transfer learning block suitable for processing 320 x 320 pixel colour images (Taylor 2023).

2.4 Implementation of AI technologies

Obviously, AI technologies can already be readily used by structure inspectors during the course of, or immediately after, a routine inspection. This might involve obtaining images for analysis using a handheld digital camera, or a camera supported with a mount, clamps or tripod (which aid in keeping the camera completely stable). The use of camera mounts is also beneficial for replicating specific angles for repeated images of a defect over time, although exact replication of parameters is virtual impossible for manual methods as discussed above. To improve efficiency, and access more difficult to reach locations, such as tunnels and bridge pylons, the use of automated camera operators has also been proposed (Peng et al. 2020; Saeed 2021)

2.4.1 Unmanned Aerial Vehicles (UAV)

Unmanned aerial vehicles (UAVs), otherwise known as drones, are being increasingly used in research applications across many and varied fields including ecology, agriculture and defence, as well as engineering (Peng et al. 2020; Abdelmalek et al. 2022; Sophie, Marcus & Nathan 2022). Use of a UAV substantially enhances the ease of access for engineering applications, where large structures can be inspected more thoroughly and quickly than they can by a visual inspection (Szu-Pyng, Yung-Chen & Feng-Liang 2023). This approach has the advantages of reducing costs associated with inspections and improving the safety for inspectors. Most civil engineering research applications for UAVs in the current literature encompass defect detection and assessment, asset monitoring (including emergency assessment) and system mapping (Peng et al. 2020; Szu-Pyng, Yung-Chen & Feng-Liang 2023). In particular, due to the speed and accuracy of deep learning models, such as YOLO, for object detection this technology is well-suited to being deployed in conjunction with a UAV (Chen, C. et al. 2023; Szu-Pyng, Yung-Chen & Feng-Liang 2023). Kao et. al demonstrated that version 4 of YOLO was able to detect concrete cracks in a bridge structure as small as 0.22 mm wide with 92% accuracy when deployed using a UAV, indicating the feasibility of this approach (Szu-Pyng, Yung-Chen & Feng-Liang 2023).

However, there are disadvantages to drone technologies including short operating windows and their inability to perform well in inclement weather (Sophie, Marcus & Nathan 2022).

At this stage the fusion of UAV and deep learning technologies requires substantial preparation and feasibility assessment. Although increasingly reported, the quality of outputs is currently inconsistent and there are several critical steps involved in accurately assessing civil infrastructure. A case study about automatic damage detection on the Pahtajokk Bridge, a simple bridge structure, used a four (4) step framework to carry out the defect detection process using AI technologies and a UAV (Mirzazade et al. 2021). This case study provides a nice explanation of the integrated workflow required to apply both UAV and AI technologies during a civil inspection. Their framework consisted of:

- Data acquisition with an efficient flight path (for remote piloted aircraft image acquisition)
- Computer vision, training, testing and validation of neural networks
- Point cloud generation (to generate location information)
- Damage quantification.

The authors also used three (3) autonomous image recognition methods on the Pahtajokk Bridge, being:

- Masking of background in images to reduce image complexity
- Detecting potential damaged areas (images were classified as containing damage or not)
- Segmentation of potentially detected damage down to the pixel level using two models.

This case study provided an extensive report on the process required to deploy AI technologies using a UAV on a very simple structure and highlighted the areas where the technologies struggled to perform. Their conclusion was that there is potential for automatic detection of defects using AI and UAVs, but that careful preparation and model training is essential for success (Mirzazade et al. 2021).

Going forward there will increasing utilisation of UAVs and AI technologies, and the combination of technologies seems boundless. Although YOLO seems to be the current front runner, other models have been tested. Saeed (Saeed 2021) developed a model to identify cracks in concrete structures using Convolutional Neural Networks and a UAV. The model was trained, validated and transferred to a Raspberry Pi 4 incorporated on an UAV. The model presented a high level of accuracy both in training and when validated, again demonstrating that AI technologies and processes are currently available to identify a common concrete structure defect. Although the use of UAVs will likely be principally directed at obtaining images, there are other potential applications such as the incorporation of other testing technologies that are also reliant on algorithms. As an example, a feasibility assessment of bridge crack width identification on the Xiangjiang River bridge achieved greater than 90% width recognition using unmanned aerial vehicle (UAV), a SLR camera and laser rangefinder. This was

combined with neural networks and support vector machine to initially locate and then extract the cracks (Peng et al. 2020).

2.4.2 Ground based inspection units

One major limitation of UAVs is their limited flight time, and the restrictions around their potential flight paths. To circumvent these restriction other approaches to the automated acquisition of images has been investigated. Huang et al (2018) conducted a study into the identification of tunnel surface cracking and leakage defects using AI and a custom-built moving tunnel inspection unit named MTI-200a. This novel example of using a ground-based unit to acquire images demonstrates that there are multiple possible approaches to automated image acquisition that also focus on enhanced safety and difficult to access locations. This unit was also used in combination with a fully convolutional network (FCN) for image recognition (Hong-wei, Qing-tong & Dong-ming 2018)

2.4.3 Mobile Backpack /Lidar method

Jun Kang et. al. (2020) presented another novel approach to identifying concrete surface defects utilising a mobile data collection system mounted to a backpack in an indoor environment. The study aimed to leverage imagery and LIDAR sensors, deep learning algorithms, and Building Information Modelling (BIM) to build a framework for automated defect inspection. The study was organised into five (5) stages: data collection, defect detection, scene reconstruction, defect assessment, and data integration (Jun Kang et al. 2020). The CNN called ResNet-50 was used for the classification of concrete defects, with the deep learning model previously built and trained by the authors. Training of the model was extensive with 18500 patches of 256 x 256 pixel images utilised, with a ratio of 4:4:4 for classes of no defect, cracking, and spalling. The model was created using Tensorflow and implemented on a Nvidia 2070 processor. Training of the model took 75 hours, which indicates that regardless of the techniques used to obtain the images, substantial effort is still required for image processing for all approaches at this stage.

2.5 Object detection model key terminology and parameters

2.5.1 Transfer learning

Transfer learning in object detection enables developers to utilise an existing trained model as a foundation for training a new model, enhancing performance through the transfer of knowledge. Farahani and team state that it is particularly useful where there is a shortage of annotated data, and that it is suitable for use on data from different domains and distributions (Farahani et al. 2020). This project used transfer learning through the Edge Impulse portal, where the YOLOv5 model benefitted from the Microsoft COCO dataset, and the AKIDA model used the AKIDANet learning block.

2.5.2 Learning rate

Smith (2015) suggests that the learning rate for a deep learning model is the most important hyper-parameter influencing the outcome. The learning rate and loss function are used to update the model weights through stochastic gradient descent to minimise the model error. Too large or too small a learning rate can make the model diverge or converge slowly, indicating that the researcher must experiment to develop an optimal rate for their model (Smith 2015)

2.5.3 Epochs (training cycles)

An Epoch is a complete pass through the training dataset using the machine learning algorithm. The pass sees weights and biases adjusted in an effort to minimise the algorithm error or loss (Afaq & Rao 2020). An Epoch is a hyper-parameter established before training and is very important to the success of a model. Too few Epochs can result in the model underfitting the data, whereby it fails to learn the data in sufficient detail. Too many Epochs and the model can learn the training data too well, inclusive of the unwanted noise, resulting in poor performance in the real world (Afaq & Rao 2020).

2.5.4 Data capture

Capturing sufficient images for model training and testing is integral to developing an accurate prediction model. In developing their region based deep learning model for detecting multiple damage types, Cha et al. collected 297 images of defects with resolutions of 6000 x 4000 pixels, using a DSLR camera (Cha et al. 2018). The availability of adequate real-world defects to supply sufficient images for model development was a major factor in deciding the type of common defect classes used in this project.

2.5.5 Data labelling

Object detection models learn based on the information that they are provided. For this study, data images were labelled and provided to the model for training. Labelling consisted of drawing a bounding box around an object within the image and assigning the object a class (in this case crack or spall). This was done through the Edge Impulse portal, however there are many other methods available to label images. Pokhrel states that bounding boxes are typically defined by coordinates and/or width and height (Pokhrel 2020). A class is defined for each object type for detection.

2.5.6 Image augmentation

Image augmentation refers to the process of making more images out of existing images through varying techniques. For object detection it can be utilised to increase the number of images for training and validation. In the development of their crack object detection model, Cha et al. carried out horizontal

flipping to increase the quantity of their images (Cha et al. 2018). Image augmentation is considered beyond the scope of this project but could be considered for future research.

2.5.7 Feature generation

Feature generation in object detection is the process whereby object similarities are gathered from the training data for utilisation in the model development. Features can vary and may include edge profiles and key point features in images. This study using the Edge Impulse portal used automatic feature generation (Tyagi 2019).

2.5.8 Training, validation and testing data

For image uploading, sufficient images will be required of each defect to enable model development. For their automatic tunnel lining multi defect detection system using YOLO4, Zhou et al. split the available images into three(3) categories; training, validation and testing, using the ratio of 6:2:2 (Zhou, Zhang & Gong 2022). In this study we propose a similar ratio of 56:16:20 for training, validation and testing respectively.

Training data are the images that are used to train the model. Validation data are used to determine the model performance during training based on the hyper-parameters set and only has an indirect affect. Shah states that testing data are images used to test the model performance post development and provide an unbiased evaluation (Shah 2017) .

2.5.9 Model performance

There are a number of ways object detection model performance can be measured. Common to most performance metrics are the terms:

True positive (TP): An object is successfully identified
True negative (TN): There is no object and no object is identified
False positive (FP): There is no object, yet an object is identified
False negative (FN): An object is present but not identified

Performance measures for object detection can include Accuracy, Precision, Recall and the F1 Score. Calculations for these measures are detailed below:

Accuracy = (Total TP + Total TN) / (Total TP + Total FP + Total FP + Total FN)
Precision = Total TP / (Total TP + Total FP)
Recall = Total TP / (Total TP + Total FN)
F1 score = (Precision x Recall) / ((Precision + Recall)/2) (Evidently AI 2023)

For object detection, Edge Impulse utilise a confidence score threshold when determining the accuracy of predicted bounding boxes with respect to their intersection over union with the true bounding box for the object. Wenkel et al. suggest that most models set low thresholds which can result in a large number of false positives (Wenkel et al. 2021). Their study attempted to optimise model performance in relation to the confidence score. For this study, confidence thresholds of 30%, 50% and 70% were utilised across all models to determine and compare accuracy against the test images using both AKIDA and YOLOv5.

2.6 Potential limitations

There has been a lot of work carried out using various AI technologies to identify individual defects on concrete structures with high accuracy. The majority of image datasets used in AI training contain images with distinct defects and minimal background confusion. In practice, a trained inspector is presented with significant amounts of visual information when inspecting a structure. Trees, pavement, concrete, grass, gravel, bolts, guardrail, water, clouds, sunlight, and shade, to name a few visual obstacles. The inspector must absorb all this information, examine the structure, establish if any defects are present, their type, and highlight their extents. This is a skill honed by humans from birth. However, image background confusion or noise can hamper the accuracy and precision of AI model results. This is currently the main potential limitation yet to be resolved to provide an autonomous one stop defect detection system suitable for real world implementation. Figure 7 below shows just how complex a simple field image can be, showing cracking, a small not so obvious spall, vegetation, discolouration, light and shade.

Figure 3: Spall and crack image with background noise
Image courtesy of: A. Bourke

The second greatest limitation to the use of deep learning models for engineering applications is the substantial size of image datasets currently required to train the models. Multiple studies have demonstrated that small training datasets will result in poorer accuracy and potentially incorrect defect classification (Jun Kang et al. 2020; Szu-Pyng, Yung-Chen & Feng-Liang 2023). The amalgamation, labelling and sorting of these datasets prior to model training requires substantial time and expertise, time that could arguably have been used more productively in simply visually inspecting the structure. Other limitations to the use of AI technologies for civil infrastructure inspections, such as cost and computer processing power, are also important, and have been outlined in detail above.

2.7 Knowledge gap

It is unknown whether third-generation neuromorphic models can achieve the same level of accuracy and precision as the current second-generation models which use high powered and performing computers for the assessment of concrete bridge and culvert defects. This project plans to address this knowledge gap by directly comparing how well a third-generation neuromorphic model, AKIDA, performs in the detection of concrete cracks from a series of images compared with a well-investigated second-generation model, YOLO. Both models will be trained using the same image dataset, and all other parameters will be kept constant.

A second knowledge gap identified during this scoping review was whether third-generation systems can extend on the capabilities of the second-generation systems by more accurately identifying and classifying a wider range of concrete structure defects, such as spalls, graffiti and pavement cracking. This project will partially address this secondary question, by investigating the capability of AKIDA to detect spalls (as well as cracks) in images and compare the outcomes to data produced by YOLO using the same dataset.

Chapter 3: Methodology

3.1 Model selection

AKIDA and YOLO5 based object detection models were selected for comparison of their abilities to detect concrete bridge and culvert defects. AKIDA was to represent the neuromorphic technology, where-as YOLO5 was used as its' 2nd generation counterpart. Fortunately, BrainChip, the founder of AKIDA, have recently partnered with Edge Impulse in their platform to assist users in developing and deploying machine learning capabilities for sensor, audio and computer vision requirements (BrainChip 2022b). The Edge Impulse online platform allows users to develop computer vision models using both AKIDA and YOLO5 technology.

3.2 Object detection classes (defects) used for the models

The types of concrete structure defects are discussed earlier in this paper. From the authors experience in the inspection field, it was noted that concrete cracking and spalling were some of the commonly detected defects. To enable sufficient samples to be located and photographed for model development, it was decided to use cracking and spalling as the two (2) image classes for detection by the AKIDA and YOLO models.

3.3 Field Sampling

For use in model development and testing, images of the object classes were sourced from the field using an iPhone 14 camera. Over 30 concrete bridge and culvert structures were inspected by the author. It was determined that at least one hundred (100) images of each defect would be needed to develop the models, aligning with similar investigations such as that by Cha et al discussed earlier (Cha et al. 2018). Figure 4 below shows the variety of locations that bridge and culvert defect images were collected from, with the yellow points representing structure locations inclusive of New York, Philadelphia, the Brisbane Valley Rail Trail, Kalbar (QLD), Grafton (NSW), and Stanthorpe (NSW).

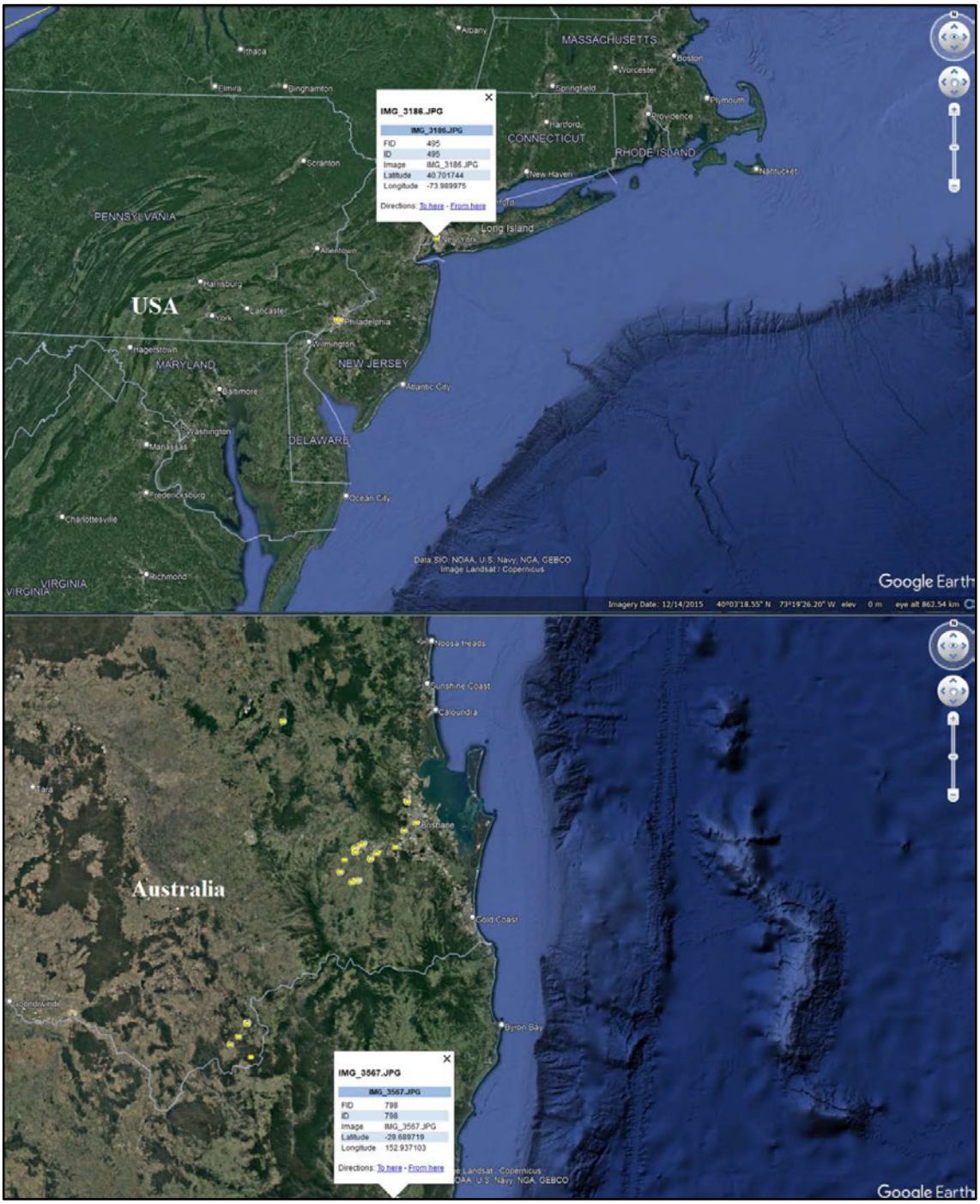


Figure 4: Location of images collected – Australia and USA
Map base layer sourced from Google Earth (https://www.google.com.au/earth/)

3.3.1 Image capture

To ensure consistency during the acquisition process and to replicate the potential close range of a drone collecting the data, the following image acquisition specifications were applied:

Resolution = *4032 x 3024 pixel (landscape or portrait)*
Camera angle = *Various*
Lighting = *Daytime, various*
Camera offset = *Approximately 1500 mm from the defect*
Background type = *Unedited – real world photographs and noise*
Location settings = *On*

Figure 5 below is an example bridge location where photographic images were collected. The yellow points represent individual photographs, with the location of IMG_3777.JPG highlighted. In total, 844 photographs were taken.

Figure 5: Bridge showing image collection locations
Map base layer sourced from Google Earth (https://www.google.com.au/earth/)

3.3.2 Image cropping

To provide a larger sample defect dataset, the 4032 x 3024 pixel images were split into tiled 512 x 512 pixel sub-images. A similar approach was taken by Cha et al (Cha et al. 2018) who used 500 x 375 pixel sub-images. This also reduced the size of the images and made them more suitable for training. A python script was developed to undertake the cropping task (refer to Appendix E for all python scripts). The script sourced original image properties on height and width to determine the number of sub images that could be created from each. The main images were then sliced into their sub-images and the sub images saved using unique file names corresponding to their sub identifier.

All relevant parent and sub-image image properties were output to a Microsoft Excel file titled, "tiled sub_image_data.xlsx", which contained the following columns:

Image_number:	The number of the parent image (844 in total)
Image_name:	File name of parent image
Latitude:	Latitude location of parent image
Longitude:	Longitude location of parent image
Sub_number:	Sub image number (numerous for each parent – different row for each)
Sub_name:	Sub image name
Sub_width:	Sub image width (typically 512 pixels)
Sub_height:	Sub image height (typically 512 pixels)
Y-Axis Coord:	Sub image starting y-coordinate on its parent image
X-Axis Coord:	Sub image starting x-coordinate on its parent image
Defect_type:	Sub image defect type (Nil if no defect present)
Defect_Quality:	Quality of defect image in the authors opinion

Each spreadsheet row represented one (1) sub image, with a reference to its parent image. As shown in Figure 7 below, the "Sub_number" column was a unique identifier for the sub image, and each unique parent "Image_number" could have multiple sub images.

3.3.3 Image sorting

Sorting the sub image tiles into folders for the respective defect types they contained was considered beneficial for later model development. To speed up the process, each parent image was imprinted with an outline grid showing the sub images and their numbers. This allowed review of all sub images on the parent image at the same time. Figure 6 shows an example of parent image 18 with viewing grids and extracted sub-image number 613.

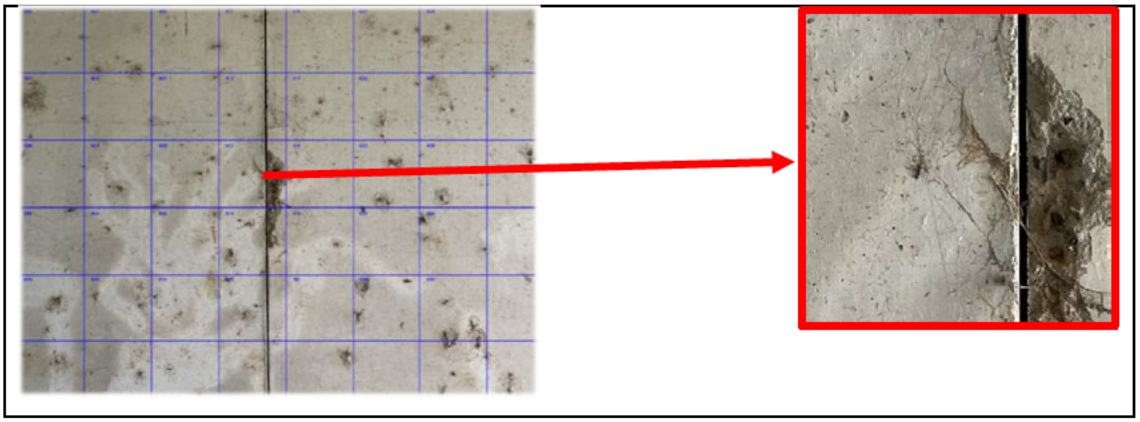


Figure 6: Example parent image #18 extract, tiles and extracted sub-image #613
Images courtesy of: A. Bourke

If a defect type was observed on a sub tile in the parent image, the Excel spreadsheet was updated under the respective row for that sub tile to capture the defect type. Figure 7 provides an example whereby a Good grade Crack was observed on tile 164. The row containing sub-tile 164 data (row 165) was updated under the 'Defect_type' column to 'Crack', and the 'Defect_Quality' column was updated to 'Good'.

	A	B	C	D	E	F	K	L
1	Image_ number	Image_name	Latitude	Longitude	Sub_n umber	Sub_ name	Defect _type	Defect_ Quality
123	4	IMG_2330.JPG	-27.6551	152.7771	122	122.jp	Crack	Good
124	4	IMG_2330.JPG	-27.6551	152.7771	123	123.jp	Crack	Good
130	4	IMG_2330.JPG	-27.6551	152.7771	129	129.jp	Crack	Good
131	4	IMG_2330.JPG	-27.6551	152.7771	130	130.jp	Crack	Good
165	5	IMG_2331.JPG	-27.6551	152.7772	164	164.jp	Crack	Good
166	5	IMG_2331.JPG	-27.6551	152.7772	165	165.jp	Crack	Good
195	6	IMG_2332.JPG	-27.6551	152.7771	194	194.jp	Crack	Good
196	6	IMG_2332.JPG	-27.6551	152.7771	195	195.jp	Crack	Good

Figure 7: Extract from image Excel datafile
Screenshot courtesy of: A. Bourke

All sub tile data shown in Figure 7 represent Good grade cracks. Classifying the sub-images was a very tedious process but was beneficial in understanding the number of defect types present for each category and which tile they were on. Only having to view the parent images and allowing for multiple sub-tile viewing expediated the process somewhat. The final Excel spreadsheet contained excellent image traceability information. Figure 8 shows sub-image 164 which represented the good quality crack image shown on row 165 of the Excel spreadsheet.

Figure 8: Cropped image #164 showing 'Good' grade Crack
Image courtesy of: A. Bourke

Once the image Excel database was updated, a python script was developed to save tiles (sub images) containing defects to a directory specific to their category. The spreadsheet defect category and quality were utilised to sort and save the images in their respective directories shown in Figure 9 below.

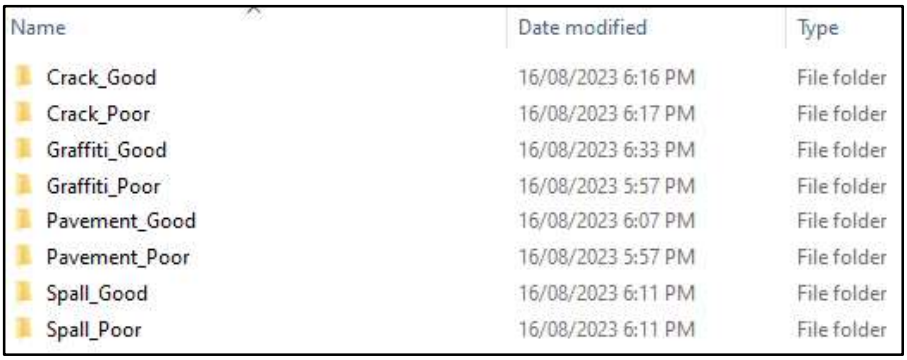


Figure 9: Example directories containing defect tile images
Screenshot courtesy of: A. Bourke

3.3.4 Developing the object detection models – YOLO and AKIDA

3.3.4.1 Model development software

Both the AKIDA and YOLO5 models were developed using the online Edge AI model development portal called 'Edge Impulse. Edge Impulse state that their platform allows the user to 'Build optimized machine learning applications that can run efficiently on any edge device'. Below is an extract showing the Edge Impulse dashboard:

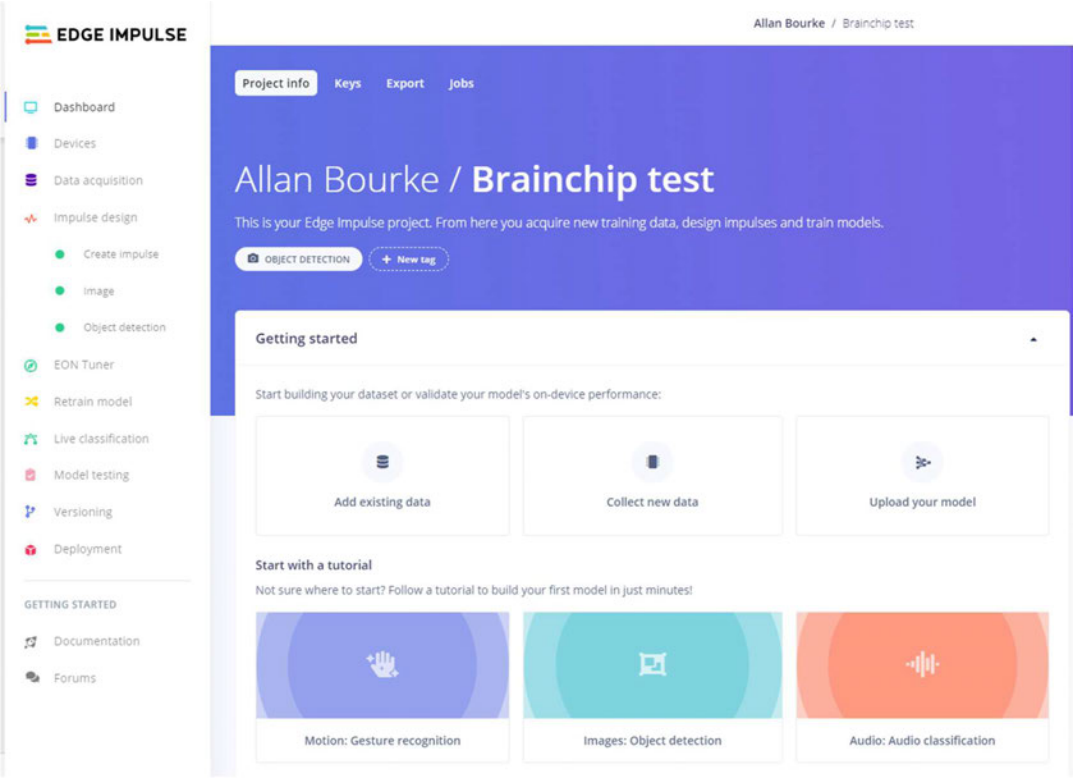


Figure 10: Edge Impulse model development portal

Source: (*Edge Impulse - Optimize AI for the edge* 2023)

3.3.4.2 Model input and optimisation

The initial model configuration was set for image object detection using a bounding box labelling method. Note as discussed earlier, the AKIDA model development using FOMO only output the bounding box centroids.

Images input into the model were initially selected and imported from the 'Crack_Good' and 'Spall_Good' datasets. One hundred (100) images from each dataset were utilised. Spall images were a limiting factor when keeping an even distribution of defects in the model, due to only 109 available spall images.

Models were also developed using crack only images, which allowed an extension of the model analysis through to 350 images. The Edge Impulse portal allowed for images to be enabled and disabled, so that different quantities of the images could be tested on the models.

All defects within the images required labelling. Images were labelled with a single bounding box as shown in Figure 11 below. During initial model exploration, it was observed that better results were being achieved when the labelling method was modified to include multiple bounding boxes for each

crack and spall within an image as demonstrated in Figure 12. This approach was utilised throughout the study.

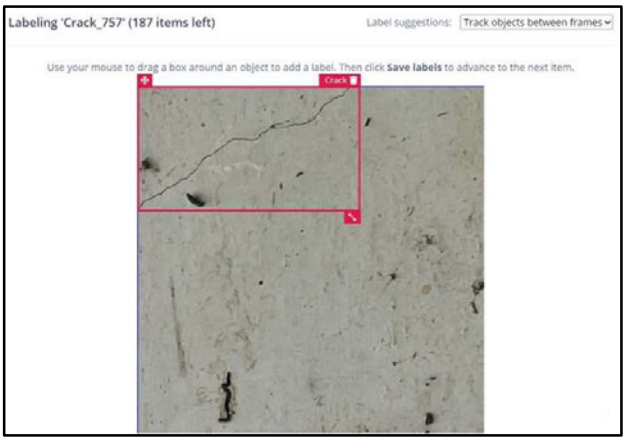


Figure 11: Initial labelling technique, select entire object
Screenshot courtesy of: A. Bourke, content of image sourced from model developed at: (*Edge Impulse - Optimize AI for the edge* 2023)

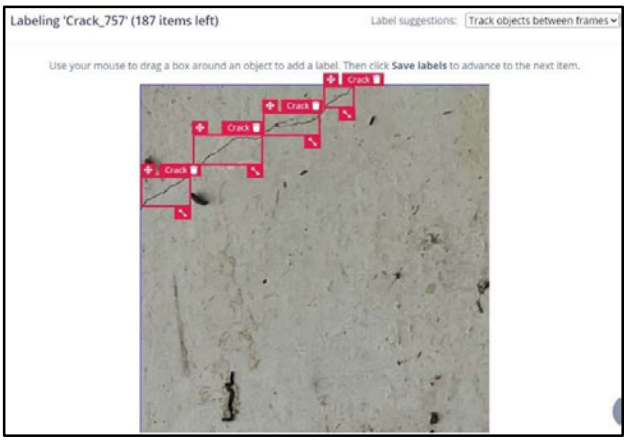


Figure 12: Optimised labelling technique, split object into small objects
Screenshot courtesy of: A. Bourke, content of image sourced from model developed at: (*Edge Impulse - Optimize AI for the edge* 2023)

For all models, 64% of images were allocated to training, 16% for validation, and 20% for model testing. The Edge Impulse developed image processing block was utilised to normalise the image data,

The custom object detection learning blocks developed by Edge Impulse and BrainChip were utilised for the YOLO5 and AKIDA models respectively. YOLOv5 community (Ultralytics 2023) and AKIDA FOMO (BrainChip 2023c) object detection models provided by Edge Impulse were subsequently used for the main training process. Both models are suitable for small datasets. Image features were generated by the Edge Impulse portal using signal processing.

Crack only models were developed for YOLOv5 and AKIDA using 100, 150, 170, 200, 350, 300 and 350 images containing cracks. A spall and crack model using 100 spall and 100 crack images was developed for analysis using YOLOv5 and AKIDA also.

In processing the models, three hundred (300) training cycles (Epochs) were used, with a varying learning rate between 0.01 and 0.001 for AKIDA to optimise testing results. The learning rate of 0.01 for the YOLO5 model could not be modified through the online portal.

To examine the effect of learning rate on a static number of images in the AKIDA model, learning rates of 0.001, 0.002, 0.003, 0.004, 0.005 and 0.01 were used in the configuration, to test the performance of an AKIDA model using 350 crack only images.

3.4 Data analysis

Both models were tested under the varying configurations. Where available through the online portal, the output accuracy, precision, and F1 scores were recorded for analysis using results from both the validation and testing data.

Precision using the validation data was examined for the YOLOv5 and AKIDA models on the crack only image models using a learning rate of 0.01. Confidence thresholds of 30%, 50% and 70%t were set to analyse the accuracy of these models using the test dataset. F1 scores were noted for all AKIDA models using the validation data; this performance metric was not available as output using the YOLOv5 model.

YOLOv5 and AKIDA models using a learning rate of 0.01 with 100 crack and 100 spall images were analysed against their accuracy using the confidence thresholds of 30%, 50% and 70% on the test data also.

3.5 Visual Analysis

To visualise and compare the models against one another, fourteen (14) images of cracks from the ‘Crack_Poor’ (crack images of poorer quality) were assembled into a power point presentation on separate slides for model testing. An animated slide show was developed showing the crack images each for two (2) seconds on the computer monitor. Using the mobile phone screen recording function and the Edge Impulse mobile phone model deployment, both models were tested and recorded live showing their ability to detect defects on the slide show images. The video MP4 files for each model were combined into a single video showing output for the same slides for each model side by side.

3.6 Data management

All image files and python scripts were stored in one drive and can be made available through a USQ approved data repository. Python scripts for image cropping which were developed by the author are also included in Appendix E.

Chapter 4: Results

4.1 Image acquisition

844 concrete bridge and culvert defects were captured and reviewed by the author, with 497 deemed suitable for inclusion in the model development. Processing of the images into sub-tiles provided 17395 512 x 512 pixel images for examination and sorting into either their defect category and quality, or nil category if no defect was present. The table below summarises the number of defect sub images that were acquired for consideration into the model. On average, approximately 2.67 defect images were acquired from each of the 497 parent images.

Table 2: Defect image count

Defect category	Count of defect tile images	Combined category count
Crack Good	369	
Crack Poor	104	473
Spall Good	107	
Spall Poor	100	207
Graffiti Good	375	
Graffiti Poor	5882	457
Pavement Good	137	
Pavement Poor	51	188
No defect	16070	
Total	17395	**1459**

4.2 Crack and spall prediction results

YOLOv5 and AKIDA models were developed using 100 crack and 100 spall images. The model accuracy using the test image dataset (40 of the 200 images) was recorded for the 30%, 50% and 70% confident thresholds. Both models produced poor results overall, less than 30% accuracy using the lowest confidence threshold. Both models produced 0% accuracy for spalls using all three (3) confidence thresholds specified. Figure 13 below summarises the overall accuracy results when predicting cracks and spalls.

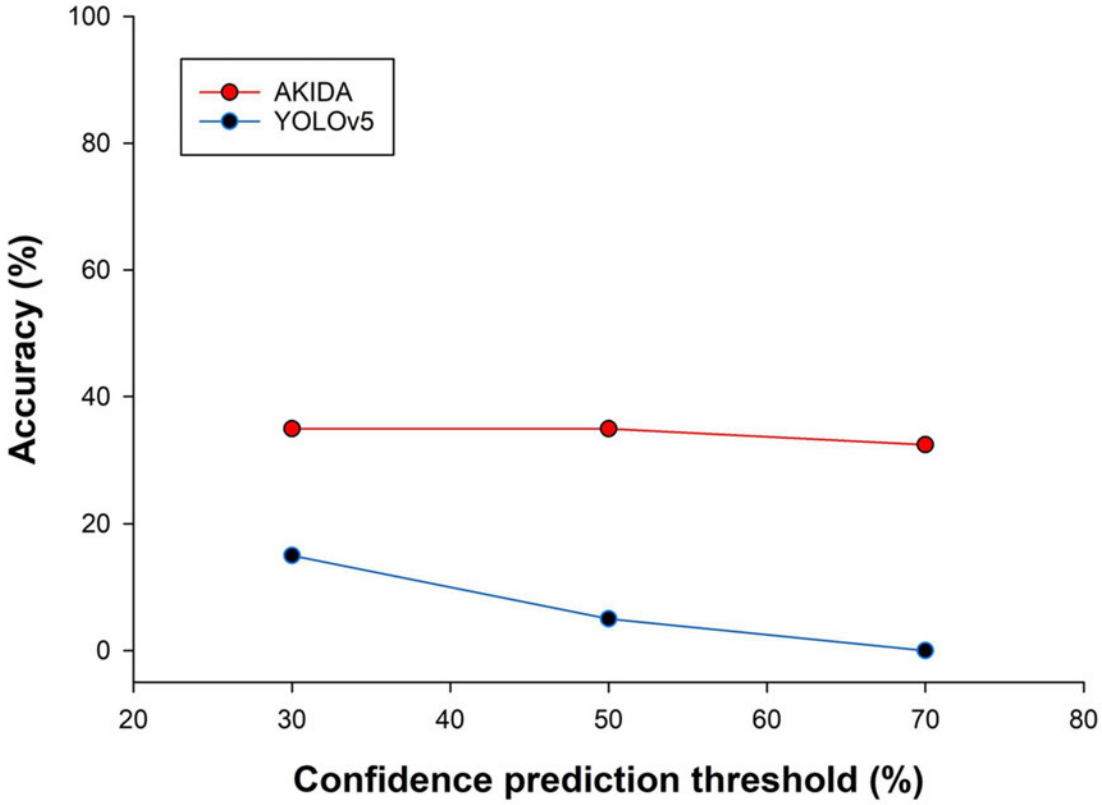


Figure 13: YOLOv5 v's AKIDA - Crack and Spall Models – trained to detect cracks and spalls.
The accuracy (%) of the AKIDA model (blue line) was higher than the YOLOv5 model (purple line) at all confidence prediction thresholds. The learning rate for the models was set at 0.01

4.3 Crack only prediction results –YOLOv5 v's AKIDA

For the development of the AKIDA models, the Edge Impulse online portal generated performance results for model accuracy, precision and F1 score using the validation data, along with accuracy for the three (3) confidence thresholds using the test data. For the YOLO model, only precision score was available on the validation data, with accuracy using the three (3) confidence thresholds on the test data available similar to AKIDA. Precision was the only common performance metric available for comparison using the validation data. Accuracy was used to compare the model results using the test data.

4.3.1 YOLOv5 v's AKIDA crack only precision results on validation data

The AKIDA model developed using 150 of the crack images produced the highest precision of all models on the validation data, with precision of 94.1%. The highest precision obtained by the YOLOv5 model was 75% using 350 crack images. Using the 150 images with YOLOv5 produced 69% precision. Table 3 below shows all model configuration combinations and their respective precision scores. F1 scores for the AKIDA model are also shown. Precision scores are further summarised in Figure 14.

Table 3: AKIDA v's YOLO 5 crack only - F1, precision and accuracy performance

Model	Learning rate	Model and # images	Validation images		Accuracy from test images given threshold		
			F1	Precision	30%	50%	70%
AKIDA	0.01	AKIDA 100	84.6	91%	75%	75%	60%
YOLOv5	0.01	YOLOv5 100	N/A	69%	15%	0%	0%
AKIDA	0.01	AKIDA 150	93	94%	67%	60%	60%
YOLOv5	0.01	YOLOv5 150	N/A	69%	13%	0%	0%
AKIDA	0.01	AKIDA 200	82.1	79%	78%	78%	63%
YOLOv5	0.01	YOLOv5 200	N/A	71%	25%	0%	0%
AKIDA	0.01	AKIDA 250	71	88%	74%	60%	48%
YOLOv5	0.01	YOLOv5 250	N/A	73%	24%	2%	0%
AKIDA	0.01	AKIDA 300	92.1	93%	82%	78%	68%
YOLOv5	0.01	YOLOv5 300	N/A	71%	23%	2%	0%
AKIDA	0.01	AKIDA 350	91.4	88%	74%	67%	50%
YOLOv5	0.01	YOLOv5 350	N/A	75%	21%	0%	0%

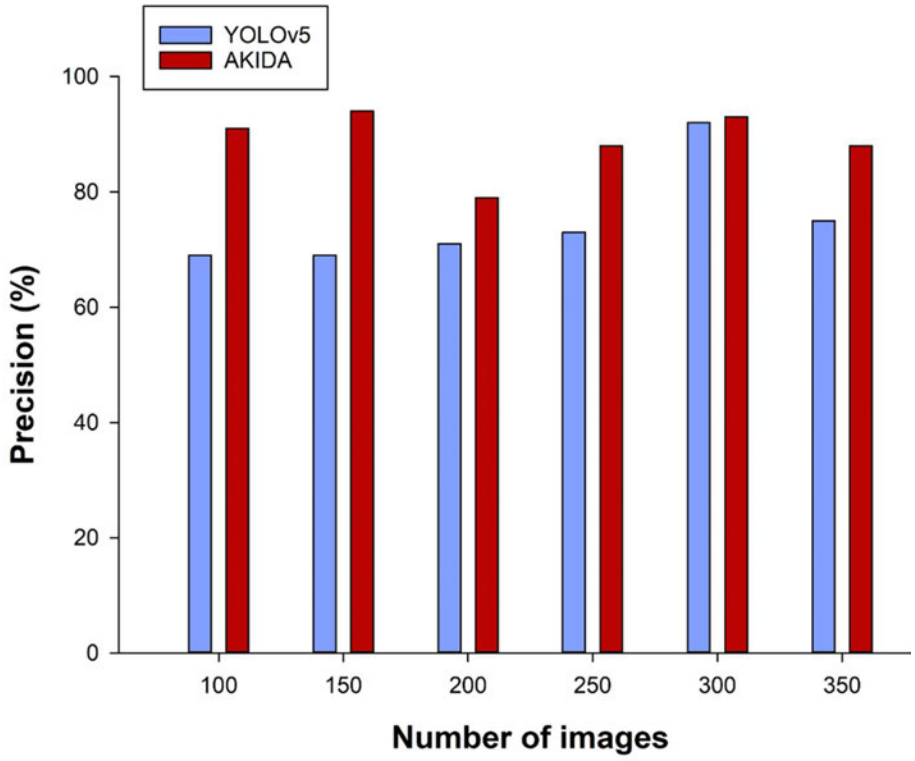


Figure 14: Two object detection models YOLOv5 v's AKIDA precision scores detecting cracks.
Performance was measured against the validation data. The precision (%) of the AKIDA Model (blue bars) was higher than the YOLOv5 model (red bars) for all image dataset sizes. The learning rate for the model was set at 0.01.

4.3.2 YOLOv5 v's AKIDA accuracy results on the test data

Table 3 above details the accuracy of the models using the 30%, 50% and 70& confidence thresholds on the test data. Again, the AKIDA model appeared have a higher accuracy under the similar model configurations. The highest accuracy achieved was 81.67% using the AKIDA model with 300 images using the 30% confidence threshold. Accuracy was slightly lower at 68%accuracy under the 70% threshold. The highest accuracy achieved by YOLOv5 was 25% using 200 images. A comparison of accuracy results is shown in Figure 15 below.

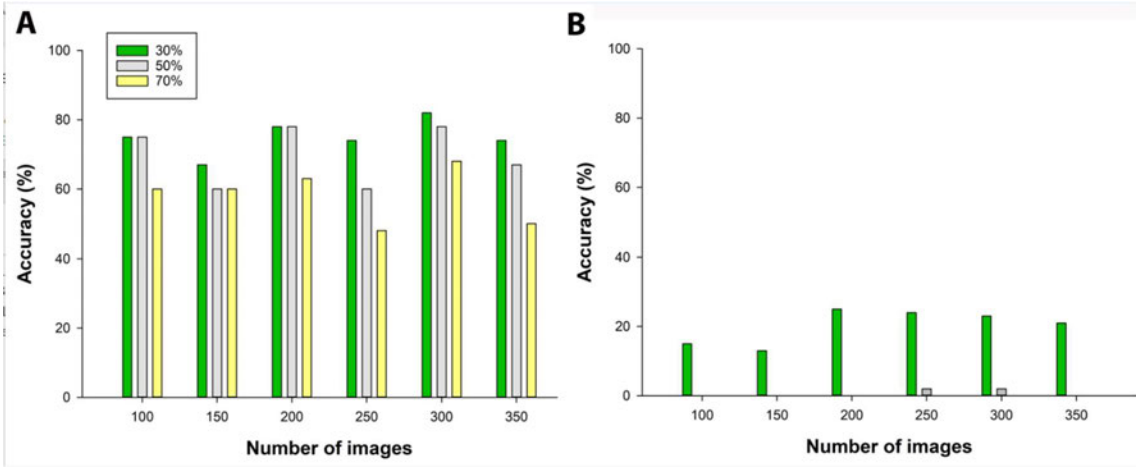


Figure 15: Two object detection models YOLOv5 v's AKIDA accuracy scores detecting cracks.
Accuracy was measured against the test data. The accuracy (%) of the AKIDA model (panel A) was higher than the YOLOv5 model (panel B) at all thresholds. The learning rate for the models was set at 0.01.

4.3.3 Learning rate effect on AKIDA model results

To enable analysis of the effect of learning rate on the model, six (6) 350 crack image only AKIDA models were developed using learning rates of 0.001, 0.002, 0.003, 0.004, 0.005 and 0.01. F1 performance on the validation data was highest for learning rate 0.01. Precision on the validation data was highest for learning rate 0.002. Accuracy results on the test images using the 30%, 50% and 70% confidence intervals were also recorded. Accuracy was highest at 80%, with a learning rate of 0.004 yielding the highest accuracy at 80% under the 30% confidence threshold. Under the 70% threshold, accuracy was highest using learning rate 0.002. Figure 16 below summarises the effect of learning rate on the AKIDA model with all other parameters static.

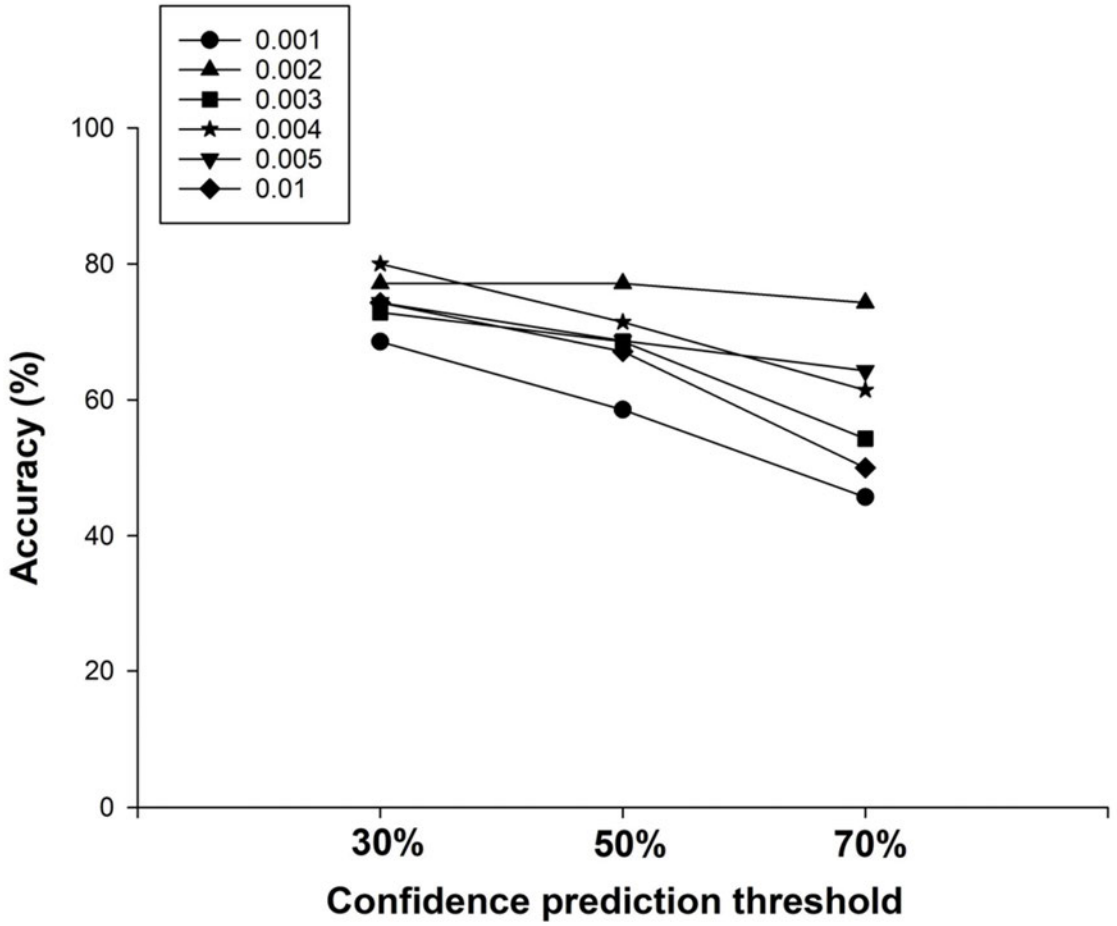


Figure 16: The effect of learning rate on an AKIDA object detection model.
350 images were used to train and test the model. Learning rates (indicated by symbols) were varied between 0.01 and 0.001.

4.3.4 Optimal AKIDA model

Whilst developing the AKIDA models for comparison against YOLOv5 using the crack only images with the static learning rate of 0.01, AKIDA was also tested for each increment of images using other learning rates to develop an optimal model. Optimisation of the AKIDA model for each image increment level is summarised in Table 4 below. Across confidence thresholds, the AKIDA 350 image model yielded the best average accuracy result of 77.1% on the test data using a learning rate of 0.002. On the validation data, this model produced a F1 result of 90.4% with 93% accuracy. Other models achieved similar performances as shown

.

Table 4: AKIDA crack only model performance optimisation

Number of model images	L-Rate	Model Accuracy on Validation	F1 on Validation	Model testing output accuracy 30% threshhold	Model testing output accuracy 50% threshhold	Model testing output accuracy 70% threshhold	Average accuracy across thresholds
100	0.01	90.8%	84.6%	75.0%	75.0%	60.0%	70.0%
100	0.03	84.5%	80.8%	60.0%	45.0%	40.0%	48.3%
100	0.05	87.0%	83.5%	65.0%	65.0%	55.0%	61.7%
100	0.09	82.0%	66.0%	63.2%	57.9%	31.6%	50.9%
100	0.1	85.0%	0.8%	68.4%	63.2%	47.4%	59.7%
100	0.12	73.0%	72.0%	57.9%	52.6%	15.8%	42.1%
150	0.01	94.1%	92.9%	66.7%	60.0%	60.0%	62.2%
150	0.03	93.6%	90.8%	63.3%	63.3%	60.0%	62.2%
150	0.05	92.4%	91.5%	66.7%	63.3%	60.0%	63.3%
170	0.01	92.0%	87.0%	58.8%	55.9%	50.0%	54.9%
170	0.03	91.0%	87.0%	73.5%	67.7%	50.0%	63.7%
170	0.05	87.0%	86.3%	75.5%	64.7%	50.0%	63.4%
170	0.1	86.0%	97.0%	50.0%	47.1%	23.5%	40.2%
200	0.008	83.0%	82.7%	67.5%	62.5%	47.5%	59.2%
200	0.01	79.1%	82.1%	77.5%	77.5%	62.5%	72.5%
200	0.03	83.0%	79.2%	70.0%	65.0%	55.0%	63.3%
250	0.002	90.3%	88.5%	66.0%	58.0%	50.0%	58.0%
250	0.005	92.6%	90.1%	72.0%	68.0%	60.0%	66.7%
250	0.008	88.5%	88.6%	70.0%	58.0%	52.0%	60.0%
250	0.01	87.6%	90.3%	74.0%	60.0%	48.0%	60.7%
300	0.001	90.0%	90.7%	80.0%	80.0%	70.0%	76.7%
300	0.002	92.6%	92.3%	76.7%	80.0%	71.7%	76.1%
300	0.005	95.4%	94.5%	76.0%	66.0%	73.3%	71.8%
300	0.01	93.1%	92.1%	81.7%	78.3%	68.3%	76.1%
350	0.001	71.7%	79.9%	68.6%	58.6%	45.7%	57.6%
350	0.002	91.1%	88.7%	77.1%	77.1%	74.3%	76.2%
350	0.002	93.0%	90.4%	82.9%	80.0%	68.6%	77.1%
350	0.003	84.8%	88.4%	72.9%	68.6%	54.3%	65.2%
350	0.004	87.1%	89.8%	80.0%	71.4%	61.4%	71.0%
350	0.005	90.5%	90.3%	74.3%	68.7%	64.3%	69.1%
350	0.01	88.1%	91.4%	74.3%	67.1%	50.0%	63.8%

4.4 Visual analysis – model deployment

Live testing of the AKIDA and YOLOv5 models was conducted on the PowerPoint slideshow containing the fourteen (14) images of cracks taken from the 'Crack_Poor' dataset. An example of the model output is shown in Figure 17 below. The author as a competent structures inspector examined the output from the models and assigned a winning model for each image prediction based pm accuracy. One (1) point was given to the winning model for each image, with points tallied to determine the best performing model overall. Results from the visual assessment are detailed in Table 5 below.

Table 5: Visual assessment - AKIDA v's YOLOv5, video analysis score for each image and total

Model	Image														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	TOTAL
AKIDA	0	0	0	0	1	0	0	0	0	0	1	1	1	0	4
YOLOv5	1	1	1	1	1	1	1	1	1	1	0	1	1	1	13

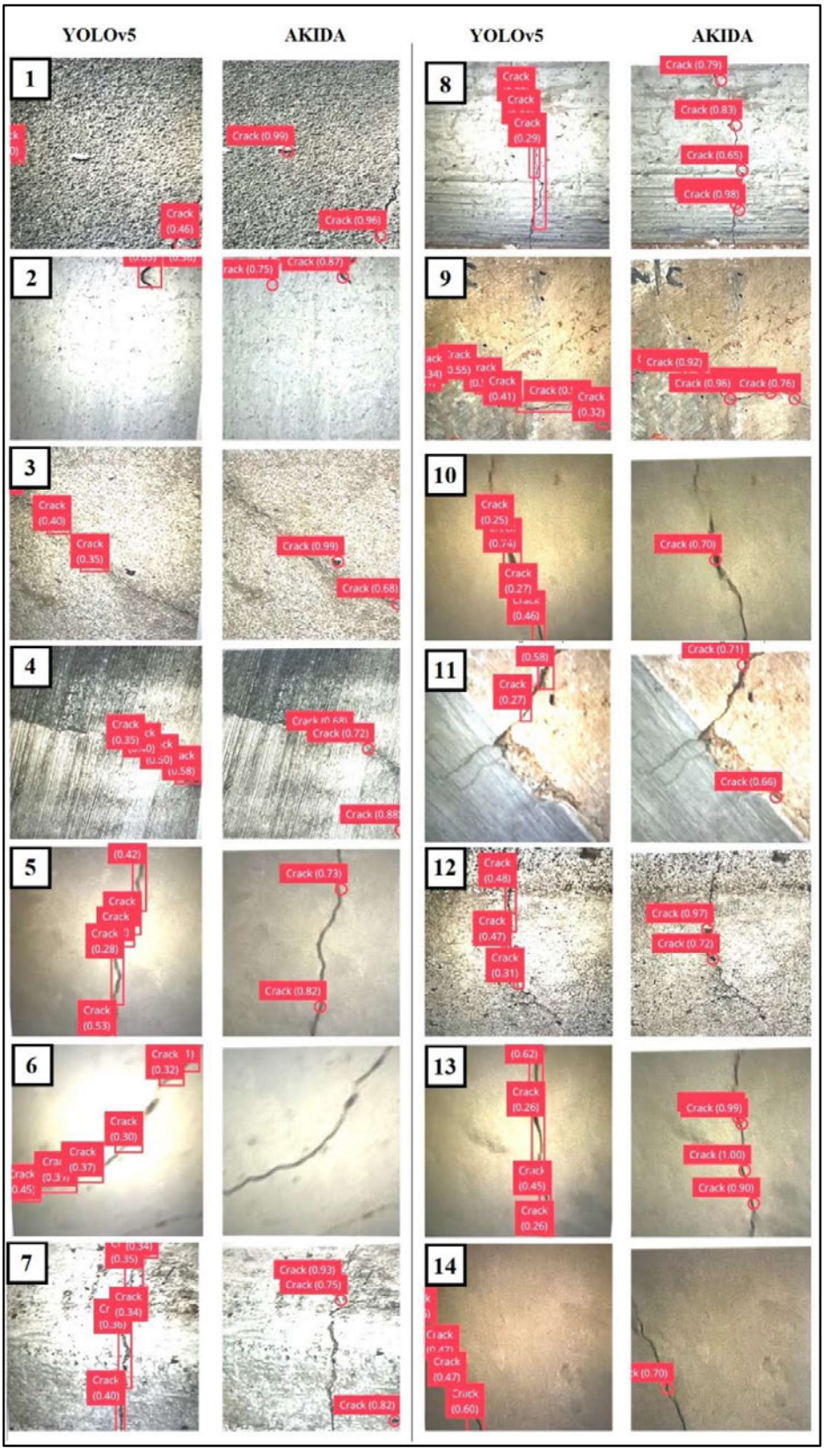


Figure 17: Video comparison of YOLO5 and AKIDA using 350 crack images for each model.
Screenshots courtesy of: A. Bourke, content sourced from models developed at: (Edge Impulse - Optimize AI for the edge 2023)

Confidence threshold levels for both models were adjusted to 25% for YOLO5 and 65% for AKIDA as these sensitivity levels appeared to be optimal when viewing the results. Lowering the confidence threshold for the AKIDA model below 65% yielded many false positives, whereas setting the YOLO5 confidence threshold above 25 % yielded very few true positives. It should be noted that given the AKIDA model was developed using the FOMO architecture, only the bounding box centroid was returned, whereas the YOLOv5 model shows the complete bounding box.

Chapter 5: Discussion

The study aim was to demonstrate that neuromorphic computing technology is a suitable novel technology to detect common defects on concrete bridge and culvert structures using object detection. Results comparing the AKIDA model to YOLOv5 are encouraging, but by no standard conclusive or robust enough to state categorically that AKIDA outperforms its counterpart. This is likely due to the lack of common model performance statistics across architectures using the Edge Impulse platform. There are sufficient grounds, however, to suggest that further investigation into the performance and benefits of neuromorphic computing for object detection would be worthwhile. The other potential benefits of neuromorphic computing not central to the considerations of this study, such as low power consumption, potential for incremental learning, improved efficiency and better security (Brainchip 2022a), also support the need for further investigation of third-generation technologies for use in civil infrastructure inspection.

Given time constraints, the acquisition of a suitable array of defect images was a limitation in developing the object detection models in the current study. Stated previously, 844 images were collected from the field, with over 30 concrete bridge and culvert structures inspected. From the collected images, 17365 sub images were assembled, with these resulting in only 107 good quality spall images and 369 good quality images of cracks. To achieve a balance of class types, the model was limited to a dataset containing 100 images of each defect, which is markedly less than used in many other studies (Sabottke & Spieler 2020; Nabizadeh & Parghi 2023). The use of transfer learning assisted in catering for this small dataset, enabling the use of predefined weights and biases to initiate the learning process. On the crack only image model, AKIDA achieved 91.4% F1 on the validation data using 350 crack images. However, visual analysis in a real-time setting that used images from outside the model learning environment suggested performance was moderately lower. Nabizadeh and Parghi utilised 1600 images of cracks in their study using YOLOv5 and achieved an F1 score of 87% (Nabizadeh & Parghi 2023). Given the difference in image dataset size between their study and the current one the similarity in F1 scores indicates that a model deployed to an AKIDA device (i.e. neuromorphic computing) has the potential to perform as well as its counterpart.

Increasing the number of crack images used in the crack only models did not appear to affect the AKIDA model F1 score. Using 100, 150 and 350 images resulted in F1 scores on the validation data of 84.6%, 93% and 91.4% respectively for crack recognition. This contrasts with the outcome for spall recognition (more discussion below). Despite the AKIDA model with 200 crack images producing an F1 score of 82.1%, when 100 of each crack and spall images were used for a combined model (i.e., 200 in total) it failed to predict one correct spall image. In the opinion of a trained inspector (A. Bourke) and as evidenced from viewing crack and spall defects in the photographs taken, spall defects could be

considered much more complex than cracks. Whilst it is important that applied models are capable of identifying the full range of potential defects on a structure, it is evident that considerably more work is required to facilitate this for defects that appear more variable in form.

The non detection of spall images was concerning given the success in detecting the crack class during the initial exploration and pre-study optimisation. The limited number of spall images available to train the models is also believed to be a major factor in the poor spall detection as mentioned above. Other studies have shown much better results using convolution neural networks and deep learning for spall detection. For their study on concrete bug-hole (small spalls) detection, Yao and team (2019) used 3200 training images and 800 validation images to develop their model, producing 96.43% accuracy on an 800 test image dataset (Yao et al. 2019). This equates to over 40 times the number of images used for model development when compared to this similar study. Further, Hoang and team (2020) achieved similar results using a combination of image processing and machine learning techniques to predict localised spalls on concrete. For their study, 600 images were utilised, which resulted in a classification accuracy of 85.32% and F1 score of 82% (Hoang 2020). It is evident from these studies that as the number of spall defect images used for model training increases, so does model accuracy and F1. Clearly, the 100 spall images utilised in this study was insufficient.

However, it is unlikely that model success is simply related to the sheer number of training images. Sample images used in Hoang and teams model are shown in their paper, with images containing spalls appearing to have relatively simple and similar backgrounds (Hoang 2020). For the current study, some defect images contained a significant amount of background noise and light variation simulating the real field environment (evidenced by Figure 17). Thus, given the variation in results between Hoang et al. and this study, it is suggested that image complexity could also affect the model performance. Secondly, performance could also be related to image resolution. Sabottke and Spieler examined model performance using varying image resolutions when detecting lesions on chest radiograph images. Their findings suggested that performance was better for 320 x 320 pixel models verses 64 x 64 pixel models (Sabottke & Spieler 2020). As the models in this study utilised images of 512 x 512 pixels, it is feasible that image resolution had no negative impact on the study. However, a direct comparison of a range of pixel densities would be required before this statement could be made definitively.

Labelling of images for model development was carried out by breaking down defects into smaller zones within the image (see Figure 12). This resulted in five (5) to six (6) times the number of labels compared to the initial labelling technique trialled during the model exploration process (see Figure 11). Bounding boxes (or centroids) subsequently covered a smaller portion of the defect. One advantage of this was that the bounding boxes, when joined, better defined the defect orientation, showing changes in direction of the crack. Direction changes are obvious from the output examples provided in Figure 17. This study finding could be particularly useful for the construction and maintenance industry who

undertake crack mapping during surveillance works. Several studies have attempted to map cracks in concrete as this information is extremely valuable when monitoring crack movement over time (Mostafa, Ahmed & Atef 2013; Jin et al. 2014). The study by Mostafa and team demonstrated that a laser scanning technique can be used for crack mapping, although its performance needed optimisation (Mostafa, Ahmed & Atef 2013). By comparison, Jin and team used digital image correlation to detect cracks in masonry walls which can open and close, but also found this methodology needs refinement (Jin et al. 2014). Thus, further studies into the use of reduced size bounding box as an alternative approach to the assessment of crack dynamics is recommended.

Initially, models were established to detect cracks and spalls in concrete structures to replicate a real world inspection and condition assessment. Ultimately, a key objective for this would be to detect all defect types, similar to those detailed in part 2 of the TMR Structure Inspection Manual (State of Queensland (Department of Transport and Main Roads) 2016b) and mentioned earlier. From the results obtained, it did not appear viable using YOLOv5 or AKIDA to continue with the assessment of combined crack and spall object detection models given the poor response to the limited number of spall images available for training. Using the lower confidence threshold on the training data did yield accuracy of up to 30%, however this was generally attributed to the overriding effect of the more successful detection of the crack class. Given the challenges with detecting spalls under the model parameters used, and considering the discussion above concerning image complexity, it does not seem likely that either of the models used would have been effective at identifying other complex defects, such as graffiti. Thus, substantial model refinement is required before real-world deployment is possible.

Due to the above concerns with image and defect complexity the study then focused on the development of crack only models. This produced more encouraging results for both YOLOv5 and AKIDA. On the validation data (16% of the total images), typically AKIDA produced a higher precision score than YOLOv5. Using the 150 crack image dataset, AKIDA performed to 94.1% precision, whereas YOLOv5 produced 69%. Precision is a measure of how repeatable an outcome is, indicating that the outcomes of the AKIDA model for the validation dataset were reliable, while the performance of the YOLOv5 model was only moderately reliable. It should be noted that precision scores were not derived for the test image dataset. This was due to limitations in the performance data metrics in the Edge Impulse portal for these models.

After training, the models were assessed against the test image dataset (20% of the total images) for accuracy, which is a measure of how correct the models were when detecting cracks. Accuracy was particularly low for the test data, yielding only 67% for AKIDA and 13% for YOLOv5 using the 30% confidence threshold. Higher thresholds resulted in even lower accuracy scores. Increasing the number of crack only images in the AKIDA model did not appear to increase accuracy on the test image dataset.

The 300 crack image models produced accuracies of 82% for AKIDA and 23% for YOLOv5 under the 30% confidence threshold condition, indicating that the AKIDA model was more accurate than the YOLOv5 model. Indeed, for all models developed, accuracy ranged from 67% to 82% for AKIDA and 13% to 25% for YOLOv5. Thus, using the confidence score thresholds of 30%, 50% and 70% on the test dataset to compare the models, FOMO deployed to AKIDA clearly outperformed YOLOv5 on accuracy (refer to Figure 15). Wenkel and team (2021) state that rarely does a paper detail the confidence score used in model evaluation, tending to only state the accuracy results, and that confidence thresholds between 5% and 25% are common (Wenkel et al. 2021). Higher thresholds result in fewer false positive predictions. Lower thresholds may result in not enough true positives. Establishing an appropriate confidence threshold for a model should be subject to the criticality of predictions and the risk and consequences of getting it wrong. For crack and spall detection in the civil engineering field for structure inspections, more false positives is probably better than no positives at all. This differs from other disciplines in the science and engineering fields where false positive diagnoses, e.g. of a chronic disease requiring long term medication, can have potentially catastrophic consequences (White & Algeri 2023). In civil infrastructure incorrect positive predictions can always be filtered out, however this is a time consuming resource intensive process that ultimately researchers are trying to eliminate in the first instance through object detection.

Examining accuracy performance from the AKIDA model using crack only images (refer to Table 4) across the validation and testing data shows that better accuracy results were obtained on the validation data as opposed to the testing data. Images for validation and testing were random but consistent across the models, although new images were introduced each time the number of images in the model was increased. Choosing the best images for inclusion in a model can be a difficult task. In this study, the number of images available to develop the models were limited from the start due to availability. Performance of a model based on a small testing dataset can be influenced considerably even with just one incorrect prediction. Mani and team (2019) suggest a method for proper evaluation of what images should be incorporated into model testing through examining coverage. Their study, which examined a number of vision detection models, demonstrated that whilst accuracy performance for their models may have been lower than expected of traditional models, the final output appeared a lot more robust (Mani et al. 2019). This could also be the case in the current study, where despite poor to average accuracy scores, both models performed well for real-time use.

When examining the model hyper-parameter ‘learning rate’ and its effect on model performance, numerous AKIDA object detection models were developed by varying the learning rate between 0.001 and 0.1, whist maintaining a constant image dataset and number of cycles. Testing of model accuracy using the confidence thresholds was conducted and the results showed a constant drop in accuracy the higher the confidence band entered, which was to be expected (refer to Figure 16). Results for the

varying learning rates appeared to be quite similar and insufficient models were developed to enable a statistical evaluation to determine if one model configuration was better than others. However, 0.002 did appear to be the optimal rate with all other hyper-parameters constant, as the accuracy curve was less steep and close to maximum when compared to the others. It is well known that learning rate can affect model performance, as it is a key component of the training algorithm (Yang & Shami 2020). For the purposes of this study, optimisation of the learning rate for the small image dataset appeared sufficient given the other model constraints. For the development and implementation of a model for the real world, it is recommended that learning rate optimisation be considered in more detail (Yang & Shami 2020). Some studies have taken learning rate optimisation further, proposing the use of a varying value based on a dynamic adjustment strategy which has produced good outcomes (Smith 2015).

In this study, whilst YOLOv5 was outperformed by AKIDA in all performance measures, it appeared through a subjective visual assessment to be more accurate and consistent in detecting concrete cracks when deployed in real time. The reason for this inconsistency is unknown, however there are a number of potential reasons. Firstly, the models were trained using a small image dataset. As discussed earlier, YOLOv5 has been shown to perform well using a large dataset. Nabizadeh and Parghi utilised 16 times the number of crack images in their model, compared to the current study, to achieve reasonable results (Nabizadeh & Parghi 2023). For this study, in the model development configuration settings, Edge Impulse suggest that both AKIDA FOMO and YOLOv5 Community learning blocks are designed to work quite well on small image datasets (*Edge Impulse - Optimize AI for the edge* 2023), yet no optimal range or minima is specified. As such, the YOLOv5 model may require additional development images to achieve better results than the present study. This is reflected by the outcome that YOLOv5 accuracy increased when more images were used in the model (refer to Figure 15 B). Secondly, the AKIDA FOMO model returned the object centroid as opposed to YOLOv5 which showed the full bounding box. FOMO is designed to only return the centroid, and as such, visual assessment was not 'like for like' across the different models. Preferably, output should have been the same (i.e., bounding box only). Thirdly, the study only touched on other model hyper-parameters (e.g. learning rate), settings and methods. Variation of any, or all, of these parameters could potentially affect model performance.

One important driver of the current study was the ability to develop a model that could be deployed for structure inspection using an UAV. Remote assessment of hard-to-access structures is the ultimate application for the deployment of a neuromorphic based model (Cha et al. 2018; Peng et al. 2020). While in this study the results suggest that there is plenty of work required before this application can be considered, the successful deployment of the AKIDA model to a mobile phone highlights that this technology could easily be adapted for drone use in the future. This finding further supports the rationale to continue this work going forward.

Finally, it is important to reflect on what went well and what could have been better optimised for this study. The key positive is that the study demonstrated that there is potential for neuromorphic computing to be used in civil engineering for concrete structure assessments. Thus, the study has achieved its aim. Further, the study has provided a carefully catalogued dataset of images that are traceable and could be used to strengthen future datasets if placed in an open access data repository. Yet the study had its limitations; evidence was quite disconnected, performance metrics did not always align (e.g. no accuracy score for YOLOv5 of validation data), deployment was through a virtual environment (as opposed to on an AKIDA1000 PCIe board or Jetson Nana device), and model customisations available through the Edge Impulse portal were not fully explored or taken advantage of. These factors all represent future improvements that could be made, given adequate time and resources. Thus, further studies comparing neuromorphic computing to traditional models should address the above limitations and consider including an assessment of other known benefits of neuromorphic computers, such as speed and energy efficiency.

Chapter 6: Conclusion

This study has demonstrated that there is potential for the use of neuromorphic computing in the Civil Engineering field for defect detection in concrete structures. Throughout the study, the AKIDA based model constantly showed performance metrics as good or better than its YOLOv5 counterpart. It is appropriate to highlight that all results for this study were achieved using the model parameters stated. Variations to learning rate, the number of training cycles, the number of images, and how images are labelled will all affect model performance. As such, the author does not suggest that results achieved are optimal to detect defects in concrete structures using a YOLOv5 or AKIDA platform. Furthermore, the functions and customisation options available through the Edge Impulse portal extend well beyond those used in models developed for this study. Researchers and users are encouraged to conduct their own exploration, configuring and testing for the development of an object detection model that best caters for their requirements.

References

Abdelmalek, B, Hafed, Z, Ahmed, K & Amine Mohammed, T 2022, 'A survey on deep learning-based identification of plant and crop diseases from UAV-based aerial images', *Cluster Computing*, vol. 26, pp. 1297 - 317.

Afaq, S & Rao, S 2020, 'Significance Of Epochs On Training A Neural Network', *International Journal of Scientific & Technology Research*, vol. 9, pp. 485-8.

Alqahtani, A & Whyte, A 2013, 'Artificial Neural Networks incorporating cost significant Items towards enhancing estimation for (life-cycle) costing of construction projects', *Australasian Journal of Construction Economics and Building*, vol. 13, pp. 51-64.

Angelova, A & McCluskey, M 2022, *Chess-playing robot breaks boy's finger at Moscow tournament*, CNN, viewed 20/05/2023, <https://edition.cnn.com/2022/07/25/europe/chess-robot-russia-boy-finger-intl-scli/index.html>.

Balayssac, J & Garnier, V 2017, *Non-destructive Testing and Evaluation of Civil Engineering Structures*.

Bilgil, A & Altun, H 2008, 'Investigation of flow resistance in smooth open channels using artificial neural networks', *Flow Measurement and Instrumentation*, vol. 19, no. 6, pp. 404-8.

Brainchip 2022a, *Edge AI: The Cloud-Free Future is Now*, BrainChip Inc., CA, USA, viewed 23 May 2022, <https://brainchip.com/wp-content/uploads/2022/05/BrainChip-w-GSA-Edge-AI-The-Cloud-Free-Future-is-Now.pdf>.

BrainChip 2022b, *BrainChip and Edge Impulse Partner to Accelerate AI/ML Deployments*, viewed 01 September 2023, <https://brainchip.com/brainchip-and-edge-impulse-partner-to-accelerate-ai-ml-deployments/>.

Brainchip 2023a, *Brainchip - Akida AKD1000*, Brainchip, viewed 20/05/2023, <https://brainchip.com/akida-neural-processor-soc/>.

Brainchip 2023b, *Akida - The global industry-standard for Edge AI*, viewed 20/05/2023, <https://brainchip.com/products/>.

BrainChip 2023c, *AkidaNet training*, viewed 16 August 2023, <https://doc.brainchipinc.com/user_guide/akida_models.html#akidanet-training>.

Bureau of Infrastructure Transport and Regional Economics (BITRE) 2021, *Yearbook 2021: Australian Infrastructure and Transport Statistics, Statistical Report,* Department of Infrastructure Transport Regional Development and Communications, BITRE, Canberra, ACT, https://www.bitre.gov.au/sites/default/files/documents/Bitre-yearbook-2021.pdf>.

Cha, Y-J, Choi, W, Suh, G, Mahmoudkhani, S & Büyüköztürk, O 2018, 'Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types', *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731-47.

Chen, C, Zheng, Z, Xu, T, Guo, S, Feng, S, Yao, W & Lan, Y 2023, 'YOLO-Based UAV Technology: A Review of the Research and Its Applications', *Drones*, vol. 7, p. 190.

Chen, L, Haoxin, Y, Xiang, Q, Shidong, Z, Peiyuang, Z, Chengwei, L, Haoyang, T, Xiu, L, Xiaohao, W & Xinghui, L 2023, 'A domain adaptation YOLOv5 model for industrial defect inspection', *Measurement*, vol. 213, p. 112725.

Christian, K, Kristina, G, Varun, K, Burcu, A & Paul, F 2015, 'A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure', *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196-210.

Dickson, B 2022, *FOMO is a TinyML neural network for real-time object detection*, viewed 25 September 2023, <https://bdtechtalks.com/2022/04/18/fomo-tinyml-object-detection/>.

Dung, C & Le Duc, A 2018, 'Autonomous concrete crack detection using deep fully convolutional neural network', *Automation in Construction*, vol. 99, pp. 52-8.

Edge Impulse - Optimize AI for the edge, 2023, Edge Impulse, viewed 01/04/2023, <https://www.edgeimpulse.com/>.

Evidently AI 2023, *A Complete Guide to Classification Metrics in Machine Learning*, viewed 11 October 2023, <https://www.evidentlyai.com/classification-metrics>.

Farahani, A, Pourshojae, B, Rasheed, KM & Arabnia, HR 2020, 'A Concise Review of Transfer Learning', *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 344-51.

Flah, M, Suleiman, AR & Nehdi, ML 2020, 'Classification and quantification of cracks in concrete structures using deep learning image-based techniques', *Cement & Concrete Composites*, vol. 114, p. 103781.

Gartner 2018, *https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders*, Gartner, Stamford, CT 06902 USA, viewed 23 May 2022, <https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders>.

GitHub Inc. 2023, *edgeimpulse/yolov5-training*, viewed 25 September 2023, <https://github.com/edgeimpulse/yolov5-training>.

Guido, M, Norman, H, Jens, K, Jakob, T, Paul, D, Marcel, H & Volker, R 2019, 'Framework for automated UAS-based structural condition assessment of bridges', *Automation in Construction*, vol. 97, pp. 77-95.

Hadi, S & Rigoberto, B 2018, 'Emerging artificial intelligence methods in structural engineering', *Engineering Structures*, vol. 171, pp. 170-89.

Hoang, N-D 2020, 'Image Processing-Based Spall Object Detection Using Gabor Filter, Texture Analysis, and Adaptive Moment Estimation (Adam) Optimized Logistic Regression Models', *Advances in Civil Engineering*, vol. 2020, p. 8829715.

Hoang, N-D, Quoc-Lam, N & Van-Duc, T 2018, 'Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network', *Automation in Construction*, vol. 94, pp. 203-13.

Hong-wei, H, Qing-tong, L & Dong-ming, Z 2018, 'Deep learning based image recognition for crack and leakage defects of metro shield tunnel', *Tunnelling and Underground Space Technology*, vol. 77, pp. 166-76.

Hooda, Y, Kuhar, P, Sharma, K & Verma, N 2021, 'Emerging Applications of Artificial Intelligence in Structural Engineering and Construction Industry', *Journal of Physics: Conference Series*, vol. 1950, p. 012062.

Horowitz, M 2014, '1.1 Computing's energy problem (and what we can do about it)', *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*: *Proceedings of the Digest of Technical Papers - IEEE International Solid-State Circuits Conference* pp. 10-4.

Hu, T, Zhao, J, Zheng, R, Wang, P, Li, X & Zhang, Q 2021, 'Ultrasonic based concrete defects identification via wavelet packet transform and GA-BP neural network', *PeerJ Computer Science*, vol. 7, p. e635.

Huang, J, Rathod, V, Sun, C, Zhu, M, Korattikara, A, Fathi, A, Fischer, I, Wojna, Z, Song, Y, Guadarrama, S & Murphy, K 2017, 'Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors', *Proceedings of the* pp. 3296-7.

Huu-Tai, T 2022, 'Machine learning for structural engineering: A state-of-the-art review', *Structures*, vol. 38, pp. 448-91.

IBM Cloud Education 2020a, *What is Artificial Intelligence*, IBM, viewed 13 October 2021, <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>.

IBM Cloud Education 2020b, *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?*, IBM, viewed 13 October 2021, <https://www.ibm.com/cloud/learn/deep-learning>.

IBM Cloud Education 2020c, *Deep Learning*, IBM, viewed 13 October 2021, <https://www.ibm.com/cloud/learn/deep-learning>.

Intel 2023, *Neuromorphic Computing - What Is Neuromorphic Computing*, Intel Corporation, viewed 20 May 2022, <https://www.intel.com.au/content/www/au/en/research/neuromorphic-computing.html>.

Ivanov, D, Chezhegov, A, Kiselev, M, Grunin, A & Larionov, D 2022, 'Neuromorphic artificial intelligence systems', *Frontiers in Neuroscience*, vol. 16, p. 959626.

Jin, H, Sciammarella, C, Yoshida, S & Lamberti, L 2014, 'Full-Field Displacement Measurement and Crack Mapping on Masonry Walls Using Digital Image Correlation', *Advancement of Optical Methods in Experimental Mechanics, Volume 3*: *Proceedings of the Advancement of Optical Methods in Experimental Mechanics, Volume 3*, H Jin, et al. (eds.), Springer International Publishing, pp. 187--96.

Jocher, G 2020, *YOLOv5*, viewed 25 September 2023, <https://github.com/ultralytics/yolov5>.

Jordan, MI & Mitchell, TM 2015, 'Machine learning: Trends, perspectives, and prospects', *Science*, vol. 349, no. 6245, pp. 255-60.

Jun Kang, C, Zhaoyu, S, Jimmy, W, Zhaofeng, L, Pin Siang, T, Kuan-fu, L, Xin, M & Yu-Hsing, W 2020, 'Artificial intelligence-empowered pipeline for image-based inspection of concrete structures', *Automation in Construction*, vol. 120, p. 103372.

Kim, B, Choi, S-W, Lee, d-e & Serfa Juan, R 2022, 'An Automated Image-Based Multivariant Concrete Defect Recognition Using a Convolutional Neural Network with an Integrated Pooling Module', *Sensors (Basel, Switzerland)*, vol. 22.

Kováč, L 2010, 'The 20 W sleep-walkers', *EMBO reports*, vol. 11, p. 2.

Kumar, P, Supraja, B, Swamy s, N & Kota, S 2021, 'Real-Time Concrete Damage Detection Using Deep Learning for High Rise Structures', *IEEE Access*, vol. PP, pp. 1-.

Mani, S, Sankaran, A, Tamilselvam, SG & Sethi, A 2019, 'Coverage Testing of Deep Learning Models using Dataset Characterization', *ArXiv*, vol. abs/1911.07309.

Marieb, EN & Hoehn, K 2019, *Human anatomy and Physiology*, 11th edn, Pearson Education, San Francisco.

Markets and Markets 2021, 'Global Neuromorphic Computing Market 2022', *Neuromorphic Computing Market*, https://www.marketsandmarkets.com/PressReleases/neuromorphic-chip.asp >.

Miguel, C, Javier, C-M, Francisco, C & Maria, JE 2022, 'A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends', *Expert Systems with Applications*, vol. 209, p. 118190.

Miles, C 2023, *Federal Aviation Adminstriation - Satellite Navigation - Global Positioning System (GPS)*, viewed 09 May 2023, <https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps>.

Mirzazade, A, Popescu, C, Blanksvärd, T & Täljsten, B 2021, 'Workflow for Off-Site Bridge Inspection Using Automatic Damage Detection-Case Study of the Pahtajokk Bridge', *Remote Sensing*, vol. 13, no. 14, p. 2665.

Mirzazade, A, Popescu, C, Gonzalez-Libreros, J, Blanksvärd, T, Täljsten, B & Sas, G 2023, 'Semi-autonomous inspection for concrete structures using digital models and a hybrid approach based on deep learning and photogrammetry', *Journal of Civil Structural Health Monitoring*.

Mostafa, R, Ahmed, E & Atef, F 2013, 'Automatic concrete cracks detection and mapping of terrestrial laser scan data', *NRIAG Journal of Astronomy and Geophysics*, vol. 2, no. 2, pp. 250-5.

Nabizadeh, E & Parghi, A 2023, 'Vision-based concrete crack detection using deep learning-based models', *Asian Journal of Civil Engineering*, vol. 24, no. 7, pp. 2389-403.

Olisa, S, Iloanusi, O, Chijindu, V & Ahaneku, M 2018, 'Edge Detection In Images Using Haar Wavelets, Sobel, Gabor And Laplacian Filters', *International Journal of Scientific & Technology Research*, vol. 7.

Park, K & Kim, B 2021, 'Dynamic neuromorphic architecture selection scheme for intelligent Internet of Things services', *Concurrency and Computation: Practice and Experience*, vol. n/a, no. n/a, p. e6357.

Peng, X, Zhong, X, Zhao, C, Chen, YF & Zhang, T 2020, 'The Feasibility Assessment Study of Bridge Crack Width Recognition in Images Based on Special Inspection UAV', *Advances in Civil Engineering*, vol. 2020, no. 8811649.

Pokhrel, S 2020, *Image Data Labelling and Annotation — Everything you need to know*, viewed 12 October 2023, <https://towardsdatascience.com/image-data-labelling-and-annotation-everything-you-need-to-know-86ede6c684b1>.

Redmon, J 2016, *YOLO: Real-Time Object Detection*, viewed 20/05/2023, <https://pjreddie.com/darknet/yolo/>.

Ren, Z, Fang, F, Yan, N & Wu, Y 2022, 'State of the Art in Defect Detection Based on Machine Vision', *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 9, pp. 661–91.

Roper, H, Kirkby, G & Baweja, D 1986, 'Long-Term Durability of Blended Cement Concretes in Structures.', *Publication SP - American Concrete Institute*, vol. 1, pp. 463-82.

Sabottke, CF & Spieler, BM 2020, 'The Effect of Image Resolution on Deep Learning in Radiography', *Radiology: Artificial Intelligence*, vol. 2, no. 1, p. e190015.

Saeed, MS 2021, 'Unmanned Aerial Vehicle for Automatic Detection of Concrete Crack using Deep Learning', *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pp. 624-8.

Schabowicz, K 2019, 'Non-Destructive Testing of Materials in Civil Engineering', *Materials*, vol. 12, p. 3237.

Schuman, CD, Kulkarni, SR, Parsa, M, Mitchell, JP, Date, P & Kay, B 2022, 'Opportunities for neuromorphic computing algorithms and applications', *Nature Computational Science*, vol. 2, no. 1, pp. 10-9.

Shah, T 2017, *About Train, Validation and Test Sets in Machine Learning*, Towards Data Science, viewed 11 October 2023, <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>.

Siddique, A, Vai, M & Pun, S 2023, 'A low cost neuromorphic learning engine based on a high performance supervised SNN learning algorithm', *Scientific Reports*, vol. 13.

Smith, LN 2015, 'Cyclical Learning Rates for Training Neural Networks', *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464-72.

Song Ee, P, Seung-Hyun, E & Haemin, J 2020, 'Concrete crack detection and quantification using deep learning and structured light', *Construction and Building Materials*, vol. 252, p. 119096.

Sophie, W, Marcus, S & Nathan, JW 2022, 'Barriers to Using UAVs in Conservation and Environmental Management: A Systematic Review', *Environmental Management*, vol. 71, pp. 1052-64.

Stallard, MM, Cameron, AM & Frank, EP 2018, 'A probabilistic model to estimate visual inspection error for metalcastings given different training and judgment types, environmental and human factors, and percent of defects', *Journal of Manufacturing Systems*, vol. 48, pp. 97-106.

State of Queensland (Department of Transport and Main Roads) 2016a, *Structures Inspection Manual Part 1: Structures Inspection Policy,* Department of Transport and Main Roads, https://www.tmr.qld.gov.au/-/media/busind/techstdpubs/Bridges-marine-and-other-structures/Structures-Inspection-Manual/SIM-Part-1.pdf?la=en>.

State of Queensland (Department of Transport and Main Roads) 2016b, *Structures Inspection Manual Part 2: Deterioration Mechanisms,* Department of Transport and Main Roads, https://www.tmr.qld.gov.au/-/media/busind/techstdpubs/Bridges-marine-and-other-structures/Structures-Inspection-Manual/SIM-Part-2.pdf?la=en>.

State of Queensland (Department of Transport and Main Roads) 2016c, *Structures Inspection Manual Part 3: Procedures,* Department of Transport and Main Roads, https://www.tmr.qld.gov.au/-/media/busind/techstdpubs/Bridges-marine-and-other-structures/Structures-Inspection-Manual/SIM-Part3.pdf?la=en>.

Syed Sahil Abbas, Z, Mohammad Samar, A, Asra, A, Nadia, K, Mamoona, A & Brian, L 2022, 'A survey of modern deep learning based object detection models', *Digital Signal Processing*, vol. 126, p. 103514.

Szu-Pyng, K, Yung-Chen, C & Feng-Liang, W 2023, 'Combining the YOLOv4 Deep Learning Model with UAV Imagery Processing Technology in the Extraction and Quantization of Cracks in Bridges', *Sensors (Basel, Switzerland)*, vol. 23.

Taylor, A 2023, *Benchmarking Akida with Edge Impulse: A Validation of Model Performance on BrainChip's Akida Platform*, viewed 25 September 2023, <https://edgeimpulse.com/blog/brainchip-akida-and-edge-impulse>.

Tyagi, D 2019, *Introduction To Feature Detection And Matching*, viewed 11 October 2023, <https://medium.com/data-breach/introduction-to-feature-detection-and-matching-65e27179885d>.

Ultralytics 2023, *Ultralytics - Welcome to the Ultralytics Community*, viewed 16 August 2023, <https://community.ultralytics.com/>.

Vanarse, A, Osseiran, A, Rassau, A & Made, Pvd 2022, 'Application of Neuromorphic Olfactory Approach for High-Accuracy Classification of Malts', *Sensors (Basel, Switzerland)*, vol. 22.

Wenkel, S, Alhazmi, K, Liiv, T, Alrshoud, S & Simon, M 2021, 'Confidence Score: The Forgotten Dimension of Object Detection Performance Evaluation', *Sensors*, vol. 21, no. 13, p. 4350.

White, T & Algeri, S 2023, 'Estimating the lifetime risk of a false positive screening test result', *PLOS ONE*, vol. 18, no. 2, p. e0281153.

Xiaoning, C, Qicai, W, Jinpeng, D, Rongling, Z & Sheng, L 2021, 'Intelligent recognition of erosion damage to concrete based on improved YOLO-v3', *Materials Letters*, vol. 302, p. 130363.

Xiongwei, W, Doyen, S & Steven, CHH 2020, 'Recent advances in deep learning for object detection', *Neurocomputing*, vol. 396, pp. 39-64.

Yang, L & Shami, A 2020, 'On hyperparameter optimization of machine learning algorithms: Theory and practice', *Neurocomputing*, vol. 415, pp. 295-316.

Yang, L, Peng, S, Nickolas, W & Yi, S 2021, 'A survey and performance evaluation of deep learning methods for small object detection', *Expert Systems with Applications*, vol. 172, p. 114602.

Yao, G, Wei, F, Yang, Y & Sun, Y 2019, 'Deep-Learning-Based Bughole Detection for Concrete Surface Image', *Advances in Civil Engineering*, vol. 2019, pp. 1-12.

Yin, W, Xiangzhen, K, Qin, F, Li, C & Junyu, F 2021, 'Modelling damage mechanisms of concrete under high confinement pressure', *International Journal of Impact Engineering*, vol. 150, p. 103815.

Zakaria, M, Karaaslan, E & Catbas, N 2022, 'Advanced bridge visual inspection using real-time machine learning in edge devices', *Advances in Bridge Engineering*, vol. 3, p. 27.

Zhang, C, Karim, M & Qin, R 2022, 'A Multitask Deep Learning Model for Parsing Bridge Elements and Segmenting Defect in Bridge Inspection Images'.

Zhang, P, Han, S, Ng, S & Wang, X-H 2018, 'Fiber-Reinforced Concrete with Application in Civil Engineering', *Advances in Civil Engineering*, vol. 2018, pp. 1-4.

Zhou, Z, Zhang, J & Gong, C 2022, 'Automatic detection method of tunnel lining multi-defects via an enhanced You Only Look Once network', *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 6, pp. 762-80.

Appendix A - Project Specification

ENG4111/4112 Research Project

Project Specification

For: Allan Bourke

Title: A review and analysis of neuromorphic computing technology to detect concrete structures defects using object detection

Major: Civil Engineering

Supervisors: Drs. Andy Nguyen and Jason Brown

Enrolment: ENG4111 - EXT S1, 2023
ENG4112 – EXT S2, 2023

Project Aim: To demonstrate that neuromorphic computing technology is a suitable novel technology to detect common defects on concrete bridge and culvert structures using object detection

Programme: **Version 1, 11 March 2023**

1. Review current concrete bridge and culvert defect inspection practices
2. Review neuromorphic computing technology use for object detection
3. Establish a methodology to incorporate neuromorphic computing through deep learning to identify concrete bridge and culvert defects
4. Inspect and photograph common bridge and culvert defects for use in the object detection system development
5. Develop and train a neuromorphic computer vision model to identify common bridge and culvert defects from video footage in real time
6. Implement the object detection model through a field trial on a device containing edge AI technology such as the BrainChip AKIDA Neural Processor or utilising a NVIDIA Jetson Nano
7. Analyse the field trial results to determine the system accuracy, effectiveness and usability
8. Consider alternatives to increase the system accuracy, effectiveness, and usability

If time and resources permit

9. Implement recommended modifications
10. Conduct a further field trial and compare the results with the previous trial
11. Consider using a drone to capture the field footage

Appendix B – Project Schedule

USQ Student Project - ENG4111/41112

Student: Allan Bourke
Timeframe: Semester 1 & 2, 2023
Student number: [redacted]
Supervisor: Dr. Andy Nguyen
Assistant Supervisor: Dr. Jason Brown

Legend	
Action	
Break/Overseas	

Project Title

A review and analysis of neuromorphic computing technology to detect concrete structures defects using object detection

Task #	Task	Week commencing: Sun, 26 Feb 2023	Sun, 5 Mar 2023	Sun, 12 Mar 2023	Sun, 19 Mar 2023	Sun, 26 Mar 2023	Sun, 2 Apr 2023	Sun, 9 Apr 2023	Sun, 16 Apr 2023	Sun, 23 Apr 2023	Sun, 30 Apr 2023	Sun, 7 May 2023	Sun, 14 May 2023	Sun, 21 May 2023	Sun, 28 May 2023	Sun, 4 Jun 2023	Sun, 11 Jun 2023	Sun, 18 Jun 2023	Sun, 25 Jun 2023	Sun, 2 Jul 2023	Sun, 9 Jul 2023	Sun, 16 Jul 2023	Sun, 23 Jul 2023	Sun, 30 Jul 2023	Sun, 6 Aug 2023	Sun, 13 Aug 2023	Sun, 20 Aug 2023	Sun, 27 Aug 2023	Sun, 3 Sep 2023	Sun, 10 Sep 2023	Sun, 17 Sep 2023	Sun, 24 Sep 2023	Sun, 1 Oct 2023	Sun, 8 Oct 2023	Sun, 15 Oct 2023	Sun, 22 Oct 2023	Sun, 29 Oct 2023	Sun, 5 Nov 2023	Sun, 12 Nov 2023	Sun, 19 Nov 2023	Sun, 26 Nov 2023	Sun, 3 Dec 2023	Sun, 10 Dec 2023
1	Fortnightly Progress Meeting																																										
2	Develop Project Specifications - Due 15 March																																										
3	Develop Research Proposal Outline																																										
4	Develop Aims, Limitations and Scope																																									
5	Prepare Progress Report - Duye 24 May 2023																																										
6	Undertake Literature Review																																										
7	Define Methodology																																										
8	Image acquisition																																										
9	Develop computer vision model using Akida																																										
10	Develop computer vision model using YOLO																																										
11	Test model/s																																										
12	Analyse results and compare to models developed by others using tradition AI																																										
13	Prepare Draft Final Report/Progress Report																																										
14	Prepare Seminar Presentation																																										
15	Deliver Seminar Presentation																																										
16	Complete Final Report																																										

Appendix C – Field Inspection Risk Assessment

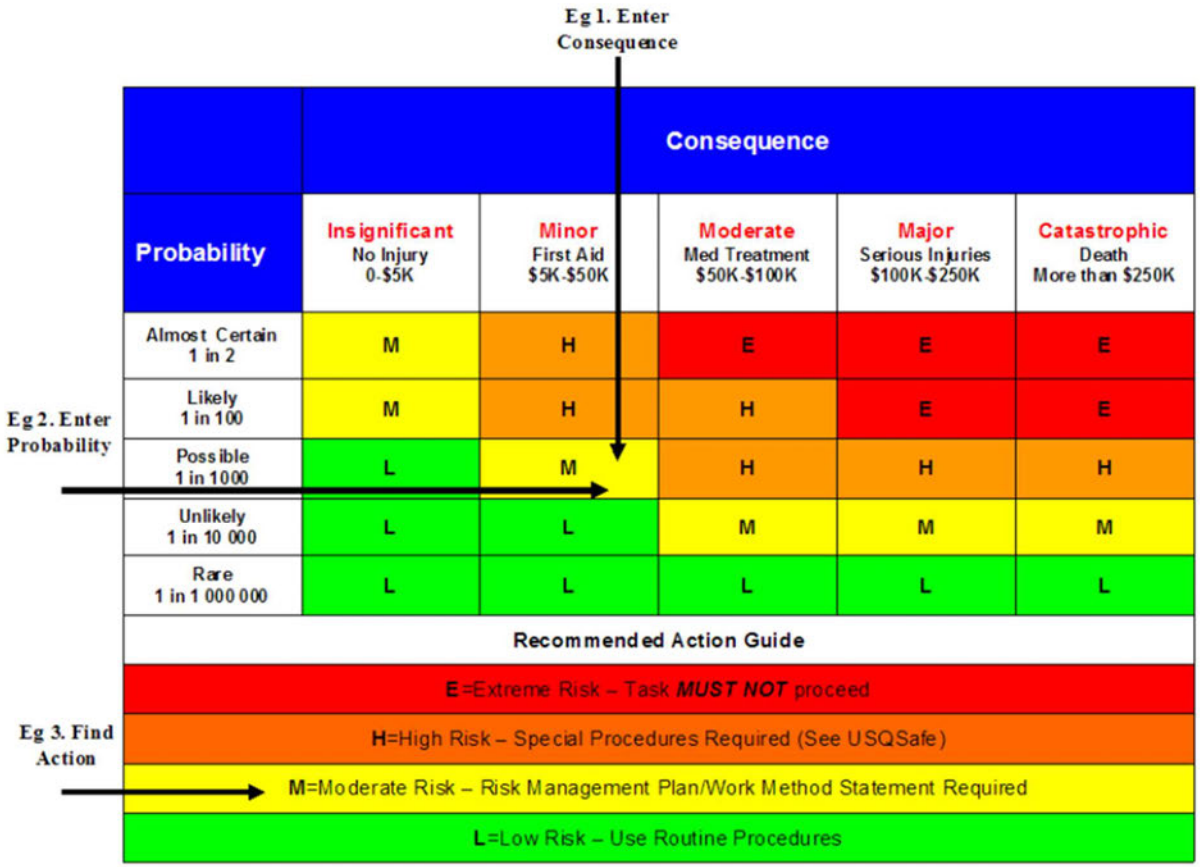
Note: This is the offline version of the Safety Risk Management System (SRMS) Risk Management Plan (RMP) and is only to be used for planning and drafting sessions, and when working in remote areas or on field activities. It must be transferred to the online SRMS at the first opportunity.

Safety Risk Management Plan – Offline Version			
Assessment Title:	A review and analysis of neuromorphic computing technology to detect concrete structure defects using object detection	Assessment Date:	25/03/2022
Workplace (Division/Faculty/Section):	USQ Faculty of Health, Engineering and Sciences	Review Date:(5 Years Max)	24/03/2023

Context			
Description:			
What is the task/event/purchase/project/procedure?	Bridge and culvert inspections		
Why is it being conducted?	Photograph concrete defects on bridge and culvert structures, such as cracks and spalling for computer vision training		
Where is it being conducted?	Various locations within Public Open Space and Road reserve		
Course code (if applicable)	ENG4111/ENG4112 - Project	Chemical name (if applicable)	
What other nominal conditions?			
Personnel involved	Dr. Andy Nguyen (Supervisor), Jason Brown (Supervisor), ALlan Bourke (USQ Student condcuting project)		
Equipment	Camera/iphone, First Aid kit, Hard Hat, High Vis vest, enclosed shoes, sun protection		
Environment	Grass/gravel/concrete areas near bridge and culvert structures		
Other	If entering and existing the roadway from informal exit, Yellow/Amber flashing light required for vehicle		
Briefly explain the procedure/process	This SRMP is for multiple inspections over Sem 1 and 2, 2022. The general process/considerations for each inspection are: Conduct pre-start meeting prior to commencing work to discuss known potential risks, the task involved involved, and to review this SRMP. If any new risks are identified, ensure they are noted, assessed, and sufficient controls are implemented. Ensure everyone is present at the prestart meeting and that they sign this safety management plan. Ensure everyone has the correct PPE. Discuss access to site. When entering and existing traffic, ensure there are sufficient gaps. Park well away from the road, where possible in a formal car park. Be aware of traffic. Ensure		

This document is uncontrolled once printed and may not be the latest version. Access the online SRMS for the latest version. Safety Risk Management Plan V1.1

	compliance with the QLD Guide to Temporary Traffic Management, TMR, Nov 2021, part 5, section 6.1 - Traffic investigations. If crossing the road, ensure there is adequate site distance. Ensure no one enters a confined space or waterway. Stay off steep embankments. Inspect structure, identify and photograph defects. Upon completion of inspection, exit site in a safe manner and leave the area clean and tidy.
Assessment Team - who is conducting the assessment?	
Assessor(s)	Dr. Andy Nguyen
Others consulted:	Jason Brown, Allan Bourke



Step 1 (cont)	Step 2	Step 2a	Step 2b	Step 3			Step 4				
Hazards: From step 1 or more if identified	*The Risk:* What can happen if exposed to the hazard without existing controls in place?	*Consequence:* What is the harm that can be caused by the hazard without existing controls in place?	*Existing Controls:* What are the existing controls that are already in place?	*Risk Assessment:* Consequence x Probability = Risk Level			*Additional controls:* Enter additional controls if required to reduce the risk level	*Risk assessment with additional controls:*			
				Probability	Risk Level	ALARP? Yes/no		Consequence	Probability	Risk Level	ALARP? Yes/no
Example											
Working in temperatures over 35° C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	catastrophic	Regular breaks, chilled water available, loose clothing, fatigue management policy.	possible	high	No	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
Slips, trips and falls	Falling over and sustaining injury	Minor	Enclosed shoes with good grip, be aware of surrounding environment, do not transverse steep embankments	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No
Unfit for work	Potential to make a poor a decision, sustain further injury,	Minor	Regular breaks, plent of sleep the night before, 0 alcohol level, adequate water, speak up if feeling unwell	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No
Live traffic	Struck be vehicle, pedestriay and cyclist	Minor	Stay well away from traffic. Wear high vis vest, hard hat and enclosed shoes. Ensure compliance with the QLD Guide to Temporary Traffic Management, TMR, Nov 2021, part 5, section 6.1 - Traffic investigations. If crossing the road, ensure there is adequate site distance.	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No
Confined space	Poor or inadequate air supply	Minor	Never enter confined or restricted space for this activity	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No
Waterway	Drowning	Minor	Never enter waterway, stay at least 1.5 m clear	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No
Steep embankments	Falling down the embankment	Minor	Do not climb or scale steep embankments around the bridges and culverts	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No

Step 1 (cont)	Step 2	Step 2a	Step 2b	Step 3			Step 4				
Hazards: From step 1 or more if identified	*The Risk:* What can happen if exposed to the hazard without existing controls in place?	*Consequence:* What is the harm that can be caused by the hazard without existing controls in place?	*Existing Controls:* What are the existing controls that are already in place?	*Risk Assessment:* Consequence x Probability = Risk Level			*Additional controls:* Enter additional controls if required to reduce the risk level	*Risk assessment with additional controls:*			
				Probability	Risk Level	ALARP? Yes/no		Consequence	Probability	Risk Level	ALARP? Yes/no
Example											
Working in temperatures over 35° C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	catastrophic	Regular breaks, chilled water available, loose clothing, fatigue management policy.	possible	high	No	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
Low light conditions	Unable to see where you are stepping, with potential to injury yourself	Minor	Inspect structures during daylight hours. If area is dark, use appropriate lighting such as head torch	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No
Flying debris from passing road users	Potential to be struck by flying object such as beer bottle	Minor	Hard hat, stay well away from the road and other road users	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No
Wildlife	Bitten, scratched	Minor	Leave wildlife alone. Wear long trousers and gaitors if walking through long grass. Always have access to a First Aid Kit and ensure someone is First Aid trained	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No
No Emergency plan	Unable to act promply in case of emergency, causing delay in receiving help or medical treatment	Minor	Be aware of your environment and agree upon a suitable evacuation/safety point In case of emergency on site: • Stop work • For injury/illness, apply appropriate First Aid • Contact Emergency Services 000	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No

Step 1 (cont)	Step 2	Step 2a	Step 2b	Step 3			Step 4				
Hazards: From step 1 or more if identified	*The Risk:* What can happen if exposed to the hazard without existing controls in place?	*Consequence:* What is the harm that can be caused by the hazard without existing controls in place?	*Existing Controls:* What are the existing controls that are already in place?	*Risk Assessment:* Consequence x Probability = Risk Level			*Additional controls:* Enter additional controls if required to reduce the risk level	*Risk assessment with additional controls:*			
				Probability	Risk Level	ALARP? Yes/no		Consequence	Probability	Risk Level	ALARP? Yes/no
Example											
Working in temperatures over 35º C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	catastrophic	Regular breaks, chilled water available, loose clothing, fatigue management policy.	possible	high	No	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
			• First Aid kit located in inspection vehicle or backpack • Be aware of surrounding traffic • Supervisor to complete incident report form within 24 hours • Minimum 1 person on site to be First Aid trained								
		Select a consequence		Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
		Select a consequence		Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
		Select a consequence		Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
				Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
				Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
				Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
				Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
				Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No

Step 5 - Action Plan (for controls not already in place)			
Additional controls:	*Resources:*	*Persons responsible:*	*Proposed implementation date:*
All controns already in place			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.

Step 6 - Approval					
Drafter's name:	Allan Bourke			Draft date:	24/03/2022
Drafter's comments:	Allan Bourke is First Aid trained and will bring First Aid Kit to site. All persons to supply own PPE. Allan Bourke to bring SRMP copy to site				
Approver's name:	Dr. Andy Nguyen		Approver's title/position:	Senior Lecturer (and project Supervisor)	
Approver's comments:	Approved				
I am satisfied that the risks are as low as reasonably practicable and that the resources required will be provided.					
Approver's signature:	[redacted]			Approval date:	Click here to enter a date.

Appendix D - Project and Personal Risk Assessment

No.	Risk	Effect	Consequence	Likelihood	Score	Control	New Score
Project Risks							
1	Poor scope definition	Failure to achieve project objectives and deliverables	Medium	Unlikely	48	Ensure sufficient time is allocated to the scope definition. Discuss scope with supervisor to ensure deliverables are achievable and relevant	40 Minor / Unlikely
2	Insufficient time or lack of commitment to the project	Failure to meet the project milestones or potentially complete the project	Medium	Possible	60	Ensure a clear and achievable project time schedule is developed and monitored. Discuss any potential time issues during weekly progress meeting	40 Minor / Unlikely
3	Lack of understanding about the project	Poor quality deliverables that may not be relevant or correct	Medium	Possible	60	If something important to the project is unclear, further research the topic. Discuss questions that may arise with academic staff and peers	40 Minor / Unlikely

No.	Risk	Effect	Consequence	Likelihood	Score	Control	New Score
4	Lack of relevant information	Inability to produce an informative inspection strategy and implementation guidelines	Medium	Possible	60	Ensure a thorough literature review is undertaken, taking key elements of the project aim and objectives into consideration. Discuss concepts and ideas with academic staff, peers, or relevant third-party stakeholders (e.g., TMR)	40 Minor / Unlikely
5	Poor communications	Poor quality or difficult to understand deliverables. Confusion amongst the project team about tasks, milestones, and general discussion	Minor	Possible	52	Ensure all communications are clear, appropriate for target audience, and well presented. Ensure weekly project meetings have a clear agenda and discussions are well documented.	40 Minor / Unlikely
6	Loss off data	Loss of time and effort, potential not being able to complete project	Medium	Possible	60	Ensure all workings are backed up regularly	40 Minor / Unlikely
Other Personal Risks							
7	Ergonomic injury from poor workstation	Personal injury	Minor	Possible	60	Conduct ergonomic self-assessment of work area Ensure all electrical equipment is in good working order, and avoid using extension leads through foot trafficked areas	40 Minor / Unlikely

No.	Risk	Effect	Consequence	Likelihood	Score	Control	New Score
9	Mental fatigue	Burnout, poor work quality, anger, anxiety, and frustration	Medium	Possible	60	Take regular rest breaks and stretch Maintain a regular and adequate sleep pattern	40 Minor / Unlikely
10	Working near live traffic/pedestrians for field inspections	Personal or third-party injury	Medium	Possible	60	High visibility vest Safety boots Hard hat Safety glasses Orange flashing light when entering and exiting traffic from the road	40 Minor / Unlikely

Appendix E – Python Scripts

#Main image processing script

```
from __future__ import print_function
from GPSPhoto import gpsphoto
import exifread
import piexif
import cv2
import os
import numpy
import xlsxwriter
from openpyxl import load_workbook

# first establish the directories we will use
Datadir = "images/" # where our main images will be located
Exceldir = "excel  processed/" # where our excel data file is
Main_image_processed = "image  processed/" # where our main images with labels and
grid lines are stored
Sub_image_processed = "sub_processed/" # where our tiled sub images are stored

# now load all of our main image names into an array called image  files
image_files= os.listdir(Datadir)

# now determine how many image files we have
number_of_main_images = len(image_files)
print("Number of main images = " + str(number_of_main_images)) # print number of
image files

# Lets set some criteria about what size sub images we wont to break our image up to
# We will specify that our sub images are 512 x 512 pixels, BUT CAN BE RESIZED LATER
IF REQUIRED
sub_image_height = 512
sub_image_width = 512
sub_image_border_size = 4 # 4 pixel border to be drawn around each sub image
color = (255, 0, 0) # color for the border around each sub image and for labels
thickness = sub_image_border_size # for drawing border around sub image
font = cv2.FONT_HERSHEY_SIMPLEX #font for sub image labels
fontScale = 1 # scale of labelling font
text_thickness = 2 # thickness of labelling text
number  of  sub  images = 0 # used to determine number of subimages in each main image
sub_image_count = 0 # set the number of sub images to zero prior to processing
main_image_count = 0 # set the number of main images to zero prior to processing

# Storage of our data details will be in an excel file
workbook_name = (Exceldir + 'sub  image  data.xlsx')
workbook = xlsxwriter.Workbook(workbook_name)
worksheet = workbook.add_worksheet()
# Setup workbook headings
worksheet_headings = ["Image  number", "Image  name","Latitude", "Longitude",
"Sub  number", "Sub  name", "Sub  width", "Sub  height", "Y-Axis Coord", "X-Axis
Coord", "Defect_type", "Defect_Quality"]

# Writing to top row
```

```
worksheet.write_row(0, 0, worksheet_headings)



# now we will process each main image
for x in range(number_of_main_images): #for each image
    # first lets learn a bit about the main image
    image = cv2.imread(Datadir + image_files[x]) # Image is a Numpy Array
    image_name = image_files[x]
    image  height = image.shape[0]
    image_width = image.shape[1]
    image_channel = image.shape[2]
    # Get the GPS data from image file
    data = gpsphoto.getGPSData(Datadir + image_files[x])  # Get image GPS data also
    # Determine other key details about image
    number_sub_images_wide = (image_width-
sub  image  border  size)/(sub  image  width+sub  image  border  size)
    number_sub_images_wide = numpy.floor(number_sub_images_wide)
    number_sub_images_high = (image_height-
sub_image_border_size)/(sub_image_height+sub_image_border_size)
    number_sub_images_high = numpy.floor(number_sub_images_high)
    total_sub_images = number_sub_images_high * number_sub_images_wide # number of
sub images in this image
    main_image_count = main_image_count+1
    # Print out all key details from our main image
    # print("Sub image number: ", number  of  sub  images)
    # print("Image name: ", image  name)
    # print("Image height: ", image  height)
    # print("Image width: ", image_width)
    # print("Image channels: ", image  channel)
    # print("Latitude: ", data['Latitude'])
    # print("Longitude: ", data['Longitude'])
    # print("Sub images across: ", number  sub  images  wide)
    # print("sub images high: ", number  sub  images  high)
    # print("Total sub images: ", total  sub  images)


    # Now draw the grid lines on our main image to create the sub images
    # Lets start with all of the vertical lines
    vertical_grid_line = 1 # first grid line will be number 1
    while vertical_grid_line <= (number_sub_images_wide+1): #extra grid line for first
column
        print(vertical_grid_line)
        # Draw vertical grid line
        start = (2 + 4 * (vertical_grid_line-1)+(sub_image_width * (vertical_grid_line-1)), 0)
        end = (2 + 4 * (vertical_grid_line-1)+(sub_image_width * (vertical_grid_line-1)),
image_height)
        cv2.line(image, start, end, color, thickness)

        vertical_grid_line += 1 # move on to next vertical grid line

    # Lets move on to the horizontal grid lines
    horizontal_grid_line = 1  # first grid line will be number 1
    while horizontal  grid  line <= (number  sub  images  high + 1):  # extra grid line for first
row
```

```
        print(horizontal_grid_line)
        # Draw horizontal grid line
        start = (0, (2 + 4 * (horizontal_grid_line-1)+sub_image_height * (horizontal_grid_line -
1)))
        end = (image_width, (2 + 4 * (horizontal_grid_line-1)+sub_image_height *
(horizontal_grid_line - 1)))
        cv2.line(image, start, end, color, thickness)

        horizontal_grid_line += 1  # move on to next horizontal grid line

    # Now lets go through our main image, extract the sub-images, save them to sub  processed
folder
    # Label all the sub images in our main image
    # Copy our main image  processed folder
    # Write all the details to our excel file as we go along

    # Column at a time for our image, inserting row at a time for our sub image in the excel file

    grid_row = 1  # starting point for first sub image
    while grid_row <= (number_sub_images_wide):
        grid_column = 1  # starting point for first sub image
        while grid_column <= (number_sub_images_high):
            sub_image_count = sub_image_count+1 # we are processing a sub image
            w_coordinate = (grid_column*sub_image_width)-
sub_image_width+(sub_image_border_size*grid_column)
            h_coordinate = (grid_row*sub_image_height)-
sub_image_height+(sub_image_border_size*grid_row)
            # print(w  coordinate)
            # print(h_coordinate)

            # Now copy (crop) our sub image
            cropped_image = image[w_coordinate:w_coordinate+sub_image_height,
h  coordinate:h  coordinate+sub  image  width]

            # Now store our sub image details in excel
            cv2.imwrite(Sub_image_processed+str(sub_image_count)+'.jpg', cropped_image)

            # Now label our main image with the sub image number
            # text
            sub_image_label = str(sub_image_count)
            # org
            org = (h_coordinate+50, w_coordinate+50) #location of text label for each sub image
            # Using cv2.putText()
            image = cv2.putText(image, sub  image  label, org, font, fontScale, color,
text_thickness, cv2.LINE_AA, False)

            # Now write our details to excel
            # New data to write:
            new_sub_image = [main_image_count, image_name, data['Latitude'],
data['Longitude'], sub_image_count, sub_image_label+".jpg", sub_image_height,
sub_image_width, w_coordinate, h_coordinate, "Nil", "Nil"]

            # Writing to row
            worksheet.write  row(sub  image  count, 0, new  sub  image)
```

```
            grid_column += 1
        grid_row +=1

    # Now show our main image with all of the grid lines
    # cv2.imshow("My Image", image)

    # Now save our processed image with labels and grid lines

    cv2.imwrite(Main_image_processed + str(main_image_count)+'.jpg', image)

    # cv2.waitKey(0)

workbook.close()
```

```
# Sub image sorting script
import pandas as pd
import cv2

Datadir = "sub_processed/" # where our main images will be located
Defectdir = "sub_sorted/" # directory where we will save images containing defects in
relevant sub directory
Excelfile = "excel_processed/sub_image_data.xlsx" # where our excel data file is

image_excel_file = pd.read_excel(Excelfile)
excel_rows =len(image_excel_file) # number of rows in excel file
print(excel_rows)
print(image_excel_file['Sub_name'].iloc[excel_rows-1])
row_counter = 0 # counter to process each row and sort images
while row_counter < excel_rows:
    # go through each row, if defect, then save to relevant defect directory
    row_defect_value = image_excel_file['Defect_type'].iloc[row_counter]
    if row_defect_value != "Nil": # we have a positive defect so save to relevant director
        defect_file_name = image_excel_file['Sub_name'].iloc[row_counter] # name of defect
sub image
        row_defect_quality = image_excel_file['Defect_Quality'].iloc[row_counter] # used to
establish subdirectory name
        save_directory_and_image =
Defectdir+row_defect_value+"_"+row_defect_quality+"/"+row_defect_value+"_"+defect_f
ile_name # this is where we will save our sub image
        image_to_sort = cv2.imread(Datadir+defect_file_name) # Image is a Numpy Array - this
is our image to relocate/sort
        cv2.imwrite(save_directory_and_image, image_to_sort) # save sorted image to correct
directory
    row_counter = row_counter+1
```