Summer 2022

Image-Based Crack Detection by Extracting Depth of the Crack Using Machine Learning

Nishat Tabassum

IMAGE-BASED CRACK DETECTION BY EXTRACTING DEPTH OF THE CRACK USING MACHINE LEARNING

by

Nishat Tabassum

Bachelor of Science
University of South Carolina, 2021

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Computer Science

College of Engineering and Computing

University of South Carolina

2022

Accepted by:

Paul Ziehl, Major Professor

Casey Cole, Major Professor

Marco Valtorta, Committee Member

Tracey L. Weldon, Vice Provost and Dean of the Graduate School

DEDICATION

I would like to dedicate this research to my friends and family who have supported me through this wonderful journey. I couldn’t have done it without their continuous and unwavering support.

ACKNOWLEDGEMENTS

First and foremost, I would like to praise Allah, the Merciful, for the blessings and strength I have received from Him throughout my research work.

I would also like to convey my gratitude to Dr. Paul Ziehl for allowing me the opportunity to focus on my research in graduate school under his guidance. I am thankful for all that Dr. Ziehl has taught me; all your advice was given to help me succeed. My journey in graduate school has been very educating and will stay with me for the rest of my career.

Dr. Casey Cole is my research advisor who has helped me through my struggles and has provided me exemplary guidance regarding my research. Without your help, I would not have been able to excel in my work. I will be forever grateful for your patience with me and taking the time to guide me during your busy schedule

Finally, I would also like to thank my fellow graduate peers and my lab members in the structural lab for motivating to be the best I can be.

ABSTRACT

Concrete structures have been a major aspect of social infrastructure since the ancient Roman times, so they have been used for many centuries. Concrete is used for the durability and support it provides to buildings and bridges. Assessing the state of these structures is important in preserving the longevity of structures and the safety of the public. Detecting cracks in their early stage allows repairs to be made without the need to replace the whole structure, so it reduces the cost. Traditional methods are slowly falling behind as technology advances and an increase in demand for a practical method of crack detection.

This study aims to review the practicality of CNN for evaluating damages from cracks autonomously. In addition, many previous methods of crack detection such as traditional manual techniques, image processing techniques, and machine learning methods are discussed. These methods will be investigated to assess the results and effectiveness of each method.

Four primary cracks and sixteen secondary cracks of varying depths were chosen to train the CNN model for binary classification of whether a crack is present. A database of images of concrete without cracks was utilized to train the CNN model to recognize the features of images with and without cracks. Multiclass CNN was trained with a dataset of known depths of cracks to predict the severity of damages and cracks. Not many studies have been done on depth prediction of cracks, so the aim of this study is to suggest regression models as an effective method of crack depth prediction.

Test results show that both the CNN models produced high accuracy in crack identification and damage zone classification. Therefore, machine learning is an effective method that can be used by civil engineers to monitor the well-being of concrete structures to reduce labor and increase time efficiency. In addition, the XGBoost of a regression model produced satisfactory accuracy in results for predicting the depths of cracks. This demonstrates the possibility of crack depth prediction. Predicting the depths of cracks is important in gaining insight into the health of the structure and can help determine the severity of the cracks and damage to the structures.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Concrete has been used in many infrastructures all around the world since the ancient Roman times [44]. Many structures such as roads, bridges, walls, pillars, and buildings rely on concrete for its stability and integrity. These structures can last for over 100 years if properly maintained and monitored, nevertheless, the structures eventually become vulnerable and can degrade from weather conditions, weight overload, and age. These structures are placed under stress from daily loading and unloading which gradually deteriorates the durability of the structure. Many buildings and bridges have partially collapsed in the past due to neglect and time, which have led to many injuries and deaths. In the United States, the American Society of Civil Engineers grades the average infrastructure to have a D+, and overall bridges are rated C+. One major factor that contributes to these ratings is the age of the structure. It is estimated that 40% of bridges in the United States are over 50 years old. There are approximately 56,000 bridges classified as structurally deficient, which comprises of 9.1% of total bridges in the United States; and to make matters worse, there are about 188 million trips across these compromised bridges daily [1]. So, it is imperative to have awareness of the damages and health of the structures to keep the general population safe.

Damages come in the form of cracks that can range from hairline to larger cracks that can become a health hazard. This does not mean every crack should raise major

concerns, but closely monitoring the crack is the key to preventing further damages. Cracks are discontinuities in the concrete and can decrease the strength and integrity of concrete components. Unforeseen failure of concrete result as a consequence of not understanding the nature of cracks and how they form [45]. Water or corrosive chemicals can invade concrete through the cracks and cause further damage to the structural integrity. Delamination and spalling can occur because of crack deterioration which ultimately undermines the safety and serviceability of the concrete structure. Delamination is when the concrete fractures and the paste layer separate causing the slab body and concrete layer to be unbound. Spalling is when the concrete has cracked and delaminated from the substrate [46]. Crack depth is defined as the measurement of discontinuity of the crack in the concrete. The depth of the crack is the distance between the crack tip and the concrete surface [47]. This can be seen in the figure below; *D* is the distance of depth of the crack.

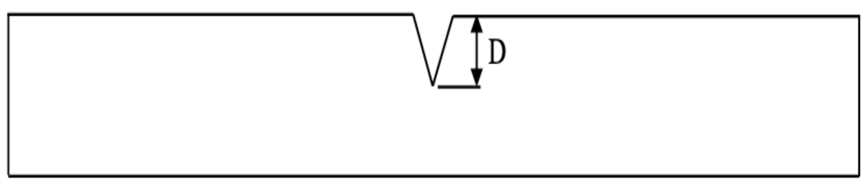


Figure 1.1 Depth Measurement of Crack

Crack depth can provide an insight on the structural integrity of the concrete. Deeper cracks indicate further damages because it can span further into the concrete, and thereby reducing the strength of the concrete. There are many types of cracks such as fatigue cracks which can be a result of cyclic short-time stress primarily on structures such as bridges that have frequent loading and unloading; impact cracks are fractures caused by an external force. Cracks can also form from a chronic overload of weight. Crack severity in civil engineering is traditionally measured by the width of the crack. Understanding the

width of cracks allows inspectors to make assumptions on the health of the structure. Cracks with widths of more than 0.3mm are considered severe and can create issues with the durability of the structure, and cracks with a width of less than 0.3mm are considered benign and do not affect the integrity of the crack [48].

If a severe crack is left untreated, it has the risk of propagating into a bigger crack and merging with adjacent cracks. Once the damage is accessed, civil engineers can repair the structure to maintain the integrity. Replacing bridges and buildings is both time consuming and costly whereas the cost to repair cracks and maintain the structures is much lower and is less time demanding. To repair cracks, first they need to be identified and categorized into separate damage zones to help prioritize the heavily damaged sections. Currently, many civil engineers must manually go out to the field to inspect the structures and later analyze the damages. Therefore, visual crack detection techniques have been one of the main focuses in computer science and can be beneficial to the field of civil engineering. Although crack identification can be obtained from a manual visual inspection, it is labor-intensive, costly, time-consuming, and often unreliable because the results are subject to the experience and skill of the inspector [2,3]. These manual inspections require specialized equipment such as thermal testing, infrared light, ultrasonic techniques, and testing concrete samples in the laboratory. They are considered invasive techniques that require multiple individuals and equipment to produce an accurate analysis of the structural integrity, so they are neither time effective nor cost efficient [4]. However, with the rapid advances in the field of computer science, applications are being developed daily that allow for non-invasive inspections using digital image analysis. There are many

new visual crack detection methods that are gaining momentum in civil engineering to improve the efficiency of monitoring the health of infrastructures.

For this research, image-based processing is utilized in which surface images are taken and processed through a machine learning algorithm that identifies cracks in the images based on its features such as length, depth, width, and location. The algorithm takes this information and classifies the severity of each crack into individual damage zones. With advancements in robotics and image capturing hardware, collection of data can become autonomous without the need of manual labor. This research demonstrates the possibility of autonomous structural inspections through the image processing and deep learning models to reduce manual labor thus reducing the risk of human error. However, there are many challenges, as the image collecting processes are prone to many factors that may hinder the accuracy of the machine learning algorithm. These factors include shadows, blemishes, noise, and illumination of the photographs [5]. To limit these and promote accuracy, deep learning methods are applied, and reports have produced results of 98% accuracy with 40,000 images in a crack detection study done by Cha et al. with Convolutional Neural Network in a report published in 2017 [6]. So, it is possible to reduce uncertainty and increase accuracy and precision of crack detection through machine learning.

There are a wide variety of other image-based methods used for crack detection each with its unique characteristics such as:

- Edge detection allows for cracks to be detected by localizing the borders of the pixels in an image. It detects the boundaries of objects and shapes within an image

through identifying the points of discontinuities in image brightness. An edge is defined as the points where there is distinct change in image brightness [12].

- Image binarization converts the pixels into grayscale images to either black or white values. In an image, dark cracks are generally categorized as black whereas lighter backgrounds produce a white value in the binarized image [8,9].
- Mathematical morphology is used as an additional process to modify crack shapes and thereby improve the identification performance [17].
- Artificial Neural Network (ANN) is a deep learning method that can be applied to detect cracks. Neural networks are programmed through the inspiration of how the human brain functions and learns [22].

Many of these methods are discussed in this study to show the traditional and most common methods of crack detection. Although they are the most common methods, new methods are being developed that involve machine learning and artificial intelligence as an alternative in hopes of creating an autonomous crack detection method.

One of the deep learning models of machine learning that is covered in this research is Convolutional Neural Network (CNN), which is used in image recognition and image processing that specifically processes pixelated data and classifies the data into specific parameters; convolution is fundamentally inserting a filter over an input. CNN has an input and output layer with multiple hidden or convolutional layers. One of the strengths of CNN is that it can handle a large complex dataset through the usage of pattern recognition. Pre-identified images of cracks with known values for depth, width, and length were used to train the CNN model to recognize patterns for cracks. For the CNN model to recognize a pattern, it must first be trained by learning the similar patterns. For example, when looking

at a person's face, features such as nose, eyes, and mouth depict that it is an image of a face; if someone does not know what nose, eyes, or mouth looks like, then they would not be able to recognize the image as a face [7]. Therefore, CNN models are trained through previously identified lines, edges, textures, and shapes to recognize similar patterns. In this research, the measured feature of a crack is trained to the CNN model to identify the presence of cracks, classify the severity, and predict the depth of concrete cracks.

1.2 OBJECTIVES AND SCOPES

1. Review previous methods used for crack detection and the effectiveness and flaws of each method.
2. Pursue the possibility of autonomous inspections for crack detection through image processing techniques utilizing machine learning algorithms and provide recommendations for further work and field use.
3. Detect the existence of concrete cracks from a novel set of images using binary classification of CNN.
4. Detect the depth level of a given crack to classify into individual categories based on the severity of the crack using multi-class classification of CNN.
5. Accurately predicting the exact depths of cracks from images using a regression model.

1.3 LAYOUT OF THESIS

This thesis consists of six chapters. In Chapter 2, a summary of current crack detection techniques and methods are given. This includes traditional manual methods, image processing techniques, and machine learning methods. Chapter 2 discusses previous

studies and findings involved in crack detection and highlights the existing gaps of knowledge.

Chapter 3 gives an overview of the research design and how it pertains to the goals and objectives to this study. It also explains the significance and the motivation of this research study.

Chapter 4 explains the experimental set up and a detailed description of the three major segments of the experiment; these are the binary classification, multiclassification, and prediction of depths in crack detection.

In Chapter 5, the results are gathered and presented from the CNN model for binary classification and multiclassification. The results are compared between the two regression models to determine the superior method.

Finally, Chapter 6 presents a summary of applications along with recommendations for future work and field use.

CHAPTER 2
LITERATURE REVIEW

2.1 INTRODUCTION

Structural integrity in concrete structures is often neglected due to the vast variety of structures that require concrete for reinforcement. Past studies on crack detection outline the significance of producing accurate results in early crack detection in the field of civil engineering. The ability to identify cracks in their early stages prevents further degradation of the structure. The urgent need to develop a reliable method in crack detection has led to variety of methods gaining momentum. There are a multitude of techniques used in crack detection that have been developed to help civil engineers monitor the health of structures. Image processing is a common method in crack detection that is still in use to this day.

Many of the traditional techniques used for concrete crack detection are manual inspections. These types of methods require an inspector to be onsite and physically testing the concrete structures with specialized equipment. One of the methods include Ultrasonic Pulse Velocity which utilizes ultrasonic pulse waves to measure the integrity of the concrete structure to indicate the presence of cracks and damages. Another manual method of crack detection is through acoustic emissions.

One of the most familiar image processing techniques used in crack detection is edge detection. Although it is relatively accurate in identifying cracks in structures, edge detection is prone to many factors that hinder the accuracy which can become challenging. Another image processing technique used in crack detection is image binarization. This method allows predictions of crack length and width utilizing binary images.

Deep learning methods have also been introduced as alternatives to traditional methods because of the low cost and practicality.

2.2 MANUAL CRACK DETECTION METHODS

Manual crack detection is performed by specialists who have experience and knowledge to correctly execute the inspection. The inspectors must be present at the inspection site to collect data for analysis of the infrastructure. Often these methods, require specialized equipment and a complex set of procedures and conditions.

2.2.1 ULTRASONIC PULSE VELOCITY (UPV)

UPV is a non-destructive evaluation technique (NDT) method for cracks, voids, discontinuations in concrete structures. This method inspects the quality and strengths of concrete without causing harm to the structure. Electronic waves are projected through the object of interest and the velocity of the ultrasonic pulse is measured to gain insight on the health of the structure. Three factors that influence the velocity are the concrete's density, elasticity, and porosity of the material tested [28]. Higher velocities indicate that the concrete is in good condition whereas lower velocities indicate that the concrete has cracks and damages [10]. One major advantage for this testing method is its simplicity, however, it requires an inspector with the skill and knowledge to operate the specialized equipment.

Many manual inspections require specialized equipment to perform the inspections. For UPV, a pulse generation circuit is utilized that consists of electronic circuit that generates pulses to send through the concrete being tested. A transducer transforms the electronic pulses into mechanical pulses having a frequency range of 50 kHz to 58 kHz [11]. Once the equipment is set up, a calibration step is needed to gauge the accuracy of the readings. A sample material with a known values and properties is used for calibration.

While this is an effective method in determining the overall health of the concrete, it is not as effective as other methods in identifying individual cracks.

There are many drawbacks to this method of crack detection. For this method to be applicable, a skilled inspector is required to operate the equipment. With a limited pool of specialized inspectors and a never-ending demand of concrete restoration, this method does not have the capabilities to inspect a variety of structures in a short amount of time. So, it is both labor-intensive and time consuming. The test results are also subject to the experience of the inspector. The inspector determines the severity of the cracks based on the results, so there is a risk of error because of human bias.

2.2.2 ACOUSTIC EMISSIONS (AE)

AE is one of the traditional methods of testing the structural integrity and crack detection in concrete. It is a type of non-destructive evaluation techniques and is useful in estimating the degree of damage on the structure [31]. AE differs from other NDT techniques in how it tests the integrity of the structure. In UPV, an ultrasonic wave is sent through the object to record the velocity to gain insight on the health of the structure. However, for AE, there is no supply of energy to the object. Instead, AE technology measures the stress waves released by the object naturally [32]. When an object is exposed to an external stimulus such as a change in pressure, load, or temperature, this results in the release of stress waves which extend to the surface of the object and can be recorded using AE sensors. This method is popular during the construction of buildings because of constant changes in loading triggers AE waves to be propagated [32]. These stress waves help inspectors gain insight on the general health of the structure and the presence of any cracks; the waves that are emitted from the structure informs inspectors if they're any

cracks. These cracks and damages may not be visible to the naked eye, so this method allows microcracks to be detected before they worsen. This is demonstrated below in the figure.

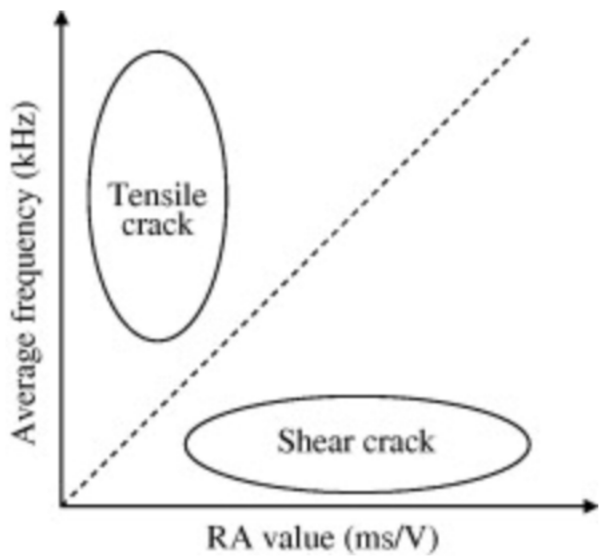


Figure 2.1 Relationship Between Average Frequency and RA Value in Acoustic Emissions

The average frequency is the average number of waves that pass a certain point per second, and the RA value is the ratio of the rise time and maximum amplitude of the AE wave [33]. Inspectors use the relationship between frequency and RA value to define whether the crack is tensile or shear. AE techniques have been used in previous studies to detect crack location, quantify the severity of damage, and distinguish between the types of cracks and damages [34,35.36]. However, there have not been many studies with AE because of the lack of practicality.

Since this technique requires AE waves to be propagated from the object, it is primarily used during construction operations. Therefore, it is not as common as other techniques. Other limitations to AE include high cost and high labor-intensity [36]. It is

also time consuming, so it does not do well to meet the high demand of an efficient crack detection and structural monitoring system.

2.3 IMAGE PROCESSING TECHNIQUES (IPT)

IPTs have been widely used to detect irregularities in concrete structures such as cracks, damages, and voids. Image processing takes an image and performs an operation to produce an enhanced image or extract controlled features. IPTs have been successful in detecting cracks; some of the most common techniques are highlighted below.

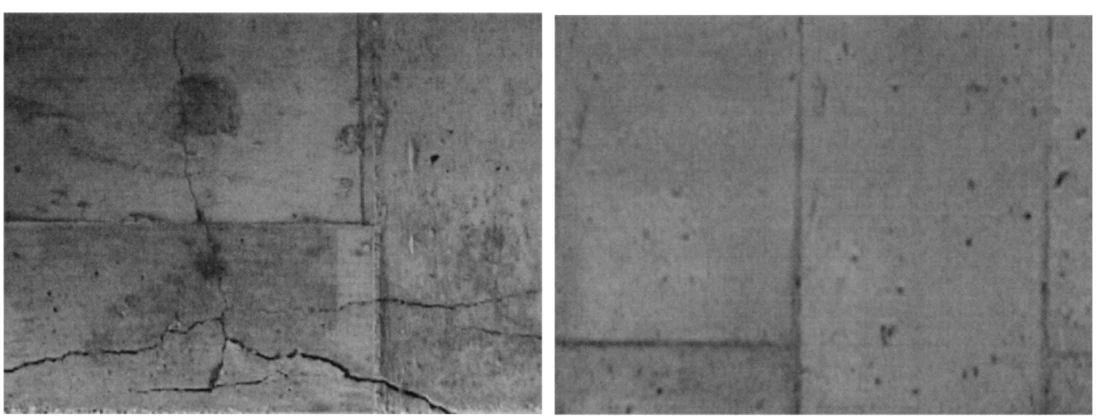
2.3.1 EDGE DETECTION

Edge detection is a method used in detecting cracks using localized pixel to detect edges within a structure in an image. These algorithm uses gaps in brightness and sharp intensity transitions on an image to measure the boundaries of objects without the need of human interference. In one study, four different edge detection algorithms were used to compare the accuracy in crack detection. These four are highlighted below:

- Sobel edge detector primarily is used for its speed and simplicity when compared to other edge detector algorithms that are more computationally complex. It is a type of spatial gradient algorithm. However, there is one major flaw to this method, Sobel is prone to image noise which may produce false positive identifications when trying to identify edges in real-world situations [12].
- Canny edge detector is more powerful and complex than the Sobel edge detector algorithm. It is convolution filter which aids in edge detection. In the smooth image, the algorithm creates an edge strength and direction at each pixel. One advantage that this algorithm has over Sobel is that it can smooth and eliminate the noise of an image by convolving with a Gaussian mass [12].

- Fast Fourier transform (FFT) is derived from Fourier transform (FT) which is a popular transform in the engineering world that uses a mathematical formula to change the domain (i.e. x-axis) of a signal from time to frequency hence the name transform. After the transformations, it is compared to a threshold to determine whether a crack is present. The ability to interconvert from spatial domain representation and frequency domain representation of images demonstrates FFT as one of the most effective tools in image processing [12].
- Fast Haar transform (FHT) is another method used to detect cracks. It is relatively new and has produced favorable results in its ability to detect cracks. Haar breaks up an image into low-frequency and high-frequency components and followed by secluding the high-frequency coefficients retrieved from the identified edge features [12].

In one study, two images were used as input for these four edge detection algorithms. The images included one with cracks on a bridge and another image of a bridge with no crack. The sample images for the bridge surface with and without cracks are shown below.



Crack image of bridge　　Image with no crack

Figure 2.2 Original Images of the Bridge Surface With and Without Cracks

These two images are inserted into each of the four algorithms to produce two edge images of the algorithms' transformation of the input images. FHT produced the highest accuracy between the four models with an accuracy of 86% [12]. It eliminated noise produced by concrete patterns therefore improving the accuracy of the results. The figure below depicts the edge images for FHT technique.

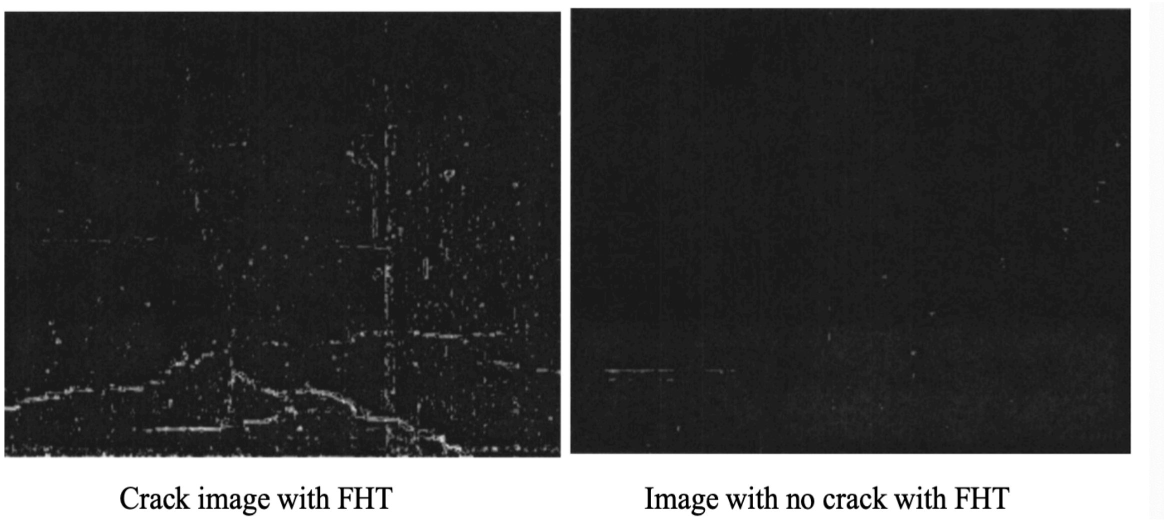
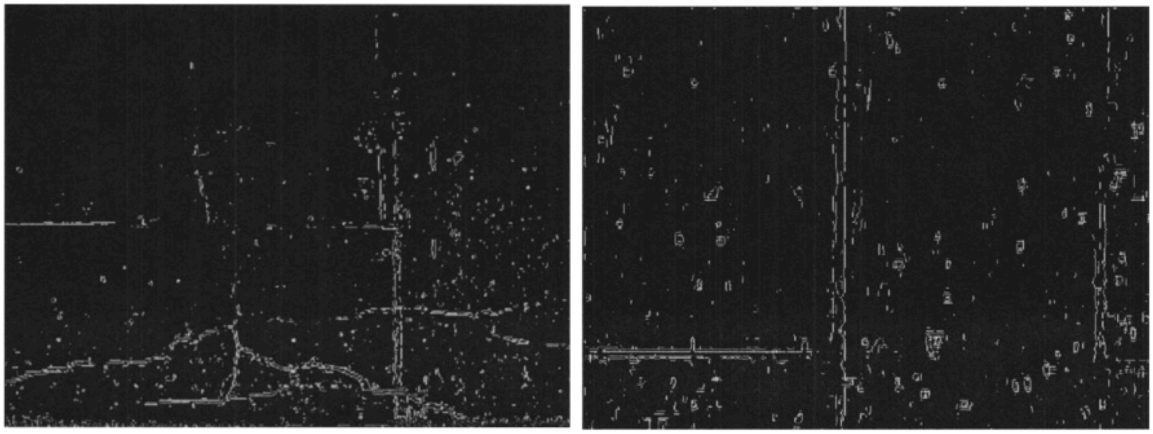


Figure 2.3 Edge Images of FHT

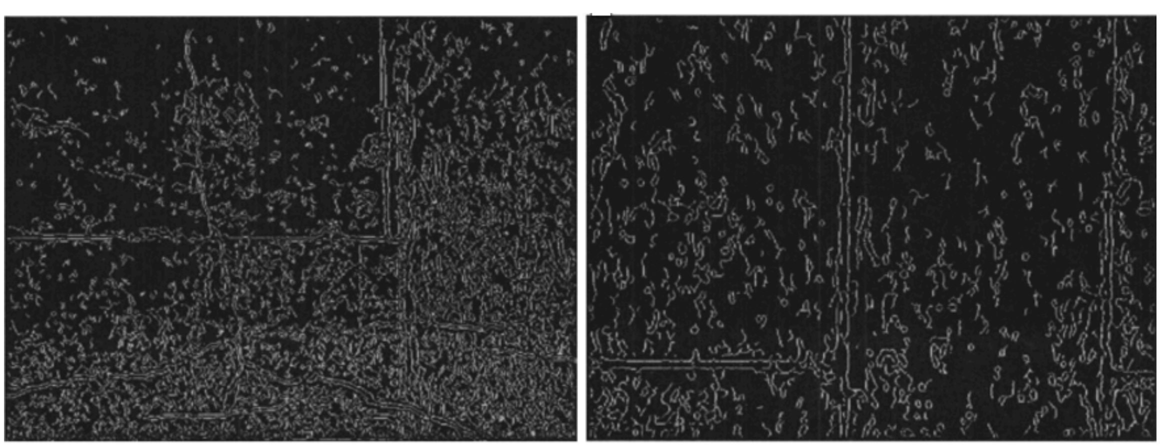
The Canny method produced results of combined accuracy of 76% and is the second most accurate model in in this study. There is a clear difference between the edge image of the image with no crack for Canny compared to the edge image of the image with no crack for FHT. Since FHT was able to better eliminate noise produced by concrete patterns and textures, it can produce results that are more accurate than the other methods [12]. However, compared to the third most accurate and the least accurate methods, the Canny edge image with no cracks has considerably less noise and distractions due to its prefilter blur that helps eliminate noise. The edge images of Canny are depicted below.



Crack image with Canny　　Image with no crack with Canny

Figure 2.4 Edge Images of Canny

The Sobel method produced the third most accurate results with combined accuracy of 68%. The Sobel method is the most simplistic in this study. It is unable to eliminate noise from the images, so this hinders the accuracy of the model and produce false positives as shown in the figure below. The main culprit to the miscalculations in the edge images is the texture of the concrete [12]. Both images have an abundance in noise which makes it hard to detect cracks accurately. The edge images of Sobel are demonstrated below.



Crack image with Sobel　　Image with no crack with Sobel

2.5 Edge Images of Sobel

The FFT method produced the lowest overall accuracy out of the four methods with an accuracy of 64%. A large factor to this low accuracy is again, because of the texture of concrete [12]. The algorithm miscalculates the texture as a blemish and results in the multitude of noise. The figure below depicts the edge images for FFT.

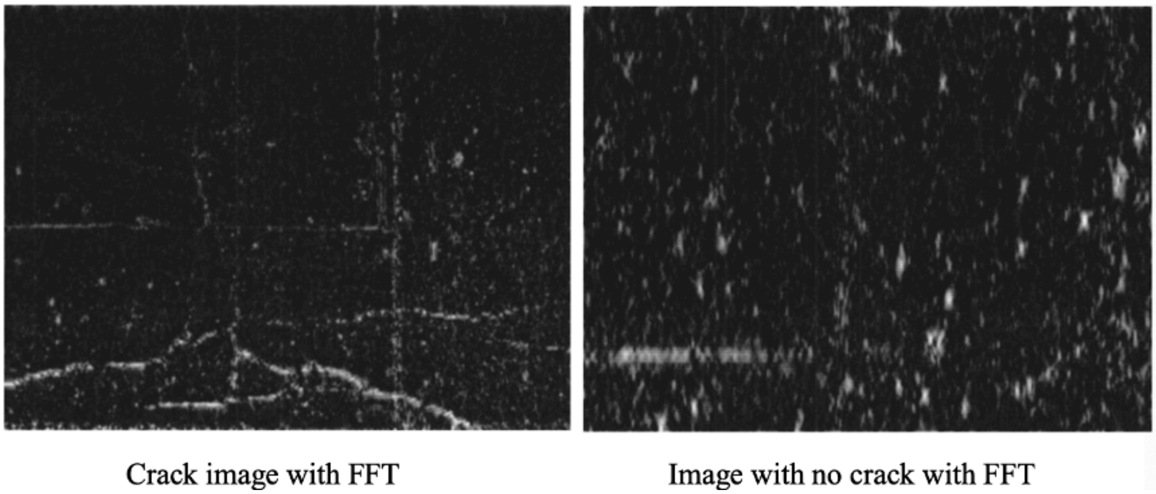


Figure 2.6 Edge Images of FFT

Edge detection is a popular method in crack detection; however, it is accompanied by many challenges. As seen in the study shown above, many edge detection methods are prone to noise produced in the image due to texture, shadows, blemishes, and illumination [12]. Shadows from natural light sources and illumination of the image contributes to many false positives in results because of the dramatic change in the brightness within the edges of a shadow which results in misidentifying a shadow as a crack. To eliminate noise and reduce miscalculations, filters can be used, and threshold values can be increased or decreased to adjust the distribution of false negatives and false positives to produce more accurate results.

2.3.2 IMAGE BINARIZATION

Image binarization is a popular image processing technique and can be used in identifying cracks in concrete. First, a color image is taken and converted into grayscale image. Then, this method converts the grayscale images to binary images. The pixel values for the grayscale image are derived from the color images which ranges from 0 to 256; these values are calculated in correlation to the weighted sum of the red, green, and blue components. These pixel values are then compared to a threshold value to determine the output of the binary image. If the pixel value is lower than the threshold, then the binary result is 1 (i.e., white). If the pixel value is higher than the threshold, then the binary result is 0 (i.e., black) [13].

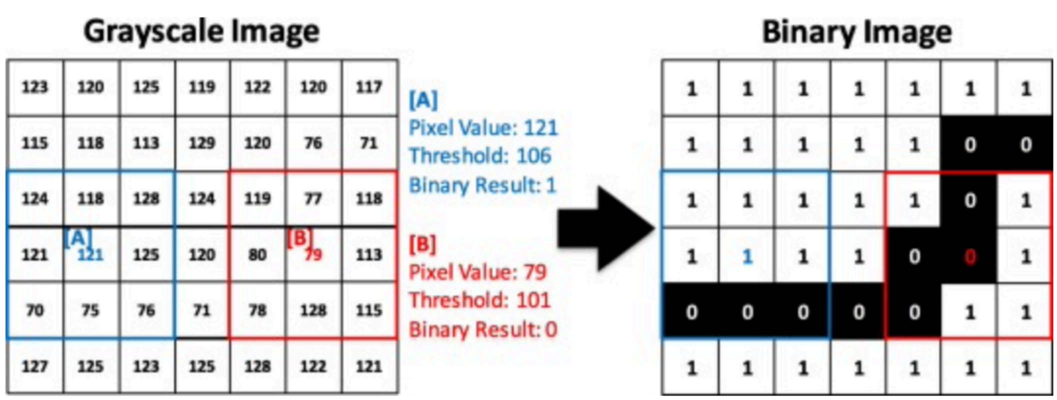


Figure 2.7 Demonstration of Image Binarization from Grayscale Image to Binary Image

In this demonstration, the window size is 3x3 for box A and box B. The threshold values in this example are derived from the weighted averages in respect to the selected window.

Setting the right parameters and threshold value is important in achieving satisfactory results [13]. There are many representation models for how the threshold value can calculated. One method by Bernsen is shown by the equation below.

$$T_{\text{Bernsen}} = \frac{Z_{\text{max}} - Z_{\text{min}}}{2}$$

(Equation 2.1)

The Z_{max} and Z_{min} values are the maximum and minimum intensities of the pixel hologram of each selected window [9]. This method is primarily used to distinguish a specific object in an image with a background with high contrast. Another method to determine the threshold was introduced by Niblack which is shown by the equation below.

$$T_{\text{Niblack}} = m + k \times s$$

(Equation 2.2)

In Niblack's representation, he focuses on using the mean (m) and standard deviation (s) to determine the threshold value where k is sensitivity [9]. This method is simple to use, however, the productivity and efficiency are significantly reduced when the background of the image has a lot of noise due to its dependency on standard deviation. Another representation for determining threshold value was developed by Sauvola which is shown in the equation below.

$$T_{\text{Sauvola}} = m \times \left\{1 - k \times \left(1 - \frac{s}{R}\right)\right\}$$

(Equation 2.3)

Sauvola modified Niblack's method to alleviate its sensitivity to the standard deviation by normalizing the standard deviation by a factor of R, the dynamic range [9]. This method is primarily used to search texts and edges from a noisy background. This method can be effective even if the background of the image is noisy, because this method considers the dynamic range. The dynamic range is defined as the contrast ratio between the darkest and brightest color tones of an image [14]. This method is useful when there is

an abundance of background noise, however, if the object and the background noise share similar sizes, it will be hard to distinguish between the two. Therefore, if there is little difference in pixel-value between the background noise and object of interest, this method would not be ideal. Wolf and Jolion developed another equation to determine the threshold value as shown below.

$$T_{Wolf} = (1 - k) \times m + k \times M + k \times \frac{s}{R} \times (m - M)$$

(Equation 2.4)

Wolf and Jolion normalized the contrast and mean to combat the challenges in Sauvola's representation of the threshold where M is the minimal pixel value of the grayscale image [9]. This method allows for dark colors to be distinguished from the background, because it considers the minimal pixel value of the entire image.

Researchers must be cautious when determining the threshold value in image binarization. There are many factors within an image that should be considered before trying to determine a method. These include the amount of background noise, the contrast in color tone in the image, and the illumination resulting in shadows. Then the window size and sensitivity value must be chosen correctly to achieve the desired results because threshold value relies heavily on these parameters [15]. The threshold value varies upon selected window size so having a larger window size results in an increase in background pixels which increases the threshold value [16]. If the window size is too large, this can lead to a misinterpretation of the values in the binary image; so, it may interpret a light crack as white in the binary image and result in a false negative. So, the key is to determine the correct parameters to increase the accuracy. If the parameters are fitted, then the crack

pixels can be localized from a grayscale image for crack detection. There is no standardized image binarization method to setting a threshold, the operators of the system must choose the best parameters according to their goals. Each representation model of threshold has a specific function, and there is not one model that accounts for all situations.

In one study, these different representations of threshold calculation were compared in crack detection for clear and unclear cracks. All the models produced errors of 11% or lower when predicting crack widths. In Wolf's method, the lengths of cracks of the unclear cracks could not be determined because the cracks themselves could not be identified; therefore, it was ineffective in determining crack length for unclear cracks. This is due to the low threshold value in this method. However, this method had the lowest number of false positive crack detections on abnormal surface textures with an abundance of background noise [9]. Other than Wolf's method, the other methods were effective in identifying both clear and unclear cracks; whereas Wolf's method was able to have the highest accuracy in determining lengths when dealing with clear cracks and produced the lowest number of false positives from noise factors such as texture, dust, or holes.

While image binarization can be used to identify cracks and even predict the length and width of cracks, it is not as effective as other methods. It is unable to predict the depth of the crack, so the analysis of the severity becomes harder without it. This method only allows for the location, length, and width to be determined. If there are small cracks or unclear cracks, this method may not indicate that they are cracks due to the parameters. As with the other methods of image processing, noisy background poses a challenge in identifying cracks in image binarization. Thus, many researchers have begun merging image binarization methods in hopes of increasing the accuracy and reducing false positive

crack detections. This is time-consuming and is computationally complex, so it may not be optimal method due to the rapid increase in need for crack monitoring systems for concrete.

2.4 DEEP LEARNING METHODS

Deep learning is a subcategory of machine learning and is a type of artificial intelligence (AI) that incorporates algorithms that are inspired from the structure of the human brain and how it functions. Machine learning is considered the fastest growing field in computer science as of today [18,19]. Neural network is a type of computer architecture that imitates how a human brain learns information by trial and error, so it is programmed to function the same way the human brain executes a given task. In recent years, deep learning has been used in speech recognition, image recognition, prediction, and object detection along with many other applications [20]. Deep learning methods for crack detection have recently been developed to overcome the challenges that image processing techniques primarily with background noise.

2.4.1 ARTIFICIAL NEURAL NETWORK (ANN)

ANN is a relatively new crack detection method in comparison to image processing techniques. However, it has had a huge impact in data science and many different fields are utilizing ANN for its accuracy, processing speed, latency, performance, and fault tolerance [21]. ANN imitates the processing operations of the human brain, however, unlike the brain, ANN does not have neurons that can connect with other neurons; it has a predetermined number of layers that are composed of nodes with an activation function [21]. There are three major classifications of layers, and they are the input, hidden, and output layers. Deep learning usually consists of multiple hidden layers which come together to form the main part of the artificial brain [22]. The input layer is where the

images are inserted to the algorithm and each hidden layer acts as a feature extraction filter. The ANN algorithm is trained for pattern recognition and has been applied in crack detection in concrete structures.

ANN has been applied to crack detection because of its success rate and accuracy. It can accommodate the limitations in image processing techniques for crack detection because it includes a preprocessing step that filters out noise to make crack detection more accurate and reduce false positive results [23]. In one study done with ANN in crack detection, 36,000 images were used to train the model, and 4,000 images were used to evaluate and produce test results. The actual measurements of one crack were recorded and the length was 396.7 mm; the minimum and maximum width were recorded to be 1.3 mm and 5.2 mm, respectively. The results from the ANN model predicts that the crack had a length of 401.3 mm and the predicted average crack width of 2.2 mm [24]. This produced error of about 1%, so it is accurate in determining the length. Furthermore, the average crack width was also within the range of the actual measured range of crack widths. This suggests that ANN is a satisfactory technique in detecting cracks and providing a prediction of the measurements.

However, ANN is not a popular method in crack detection because it is overshadowed by CNN. This is one limitation, because there are not many studies done with ANN on crack detection. Since CNN outperforms ANN in crack detection, majority of the studies focus on developing CNN. CNN is different in that the last layer is fully connected whereas in ANN, each neuron is connected to every other neuron. Visual representations of the two models are shown below.

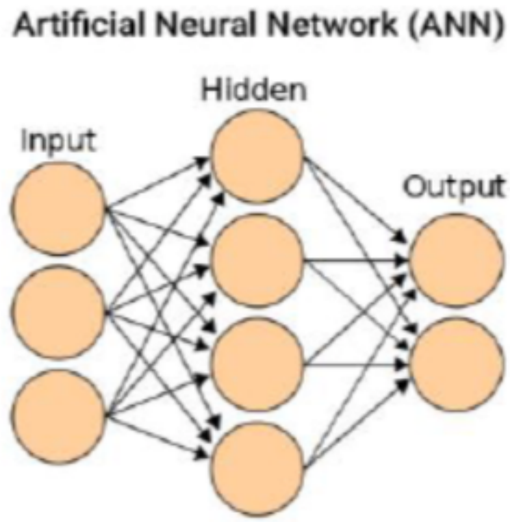


Figure 2.8 Visual Representation of ANN

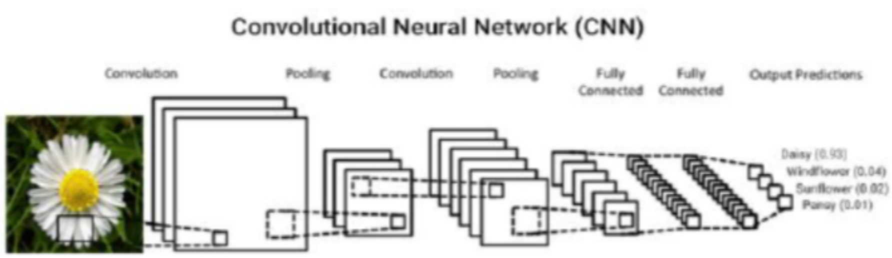


Figure 2.9 Visual representation of CNN

The fully connected network in ANN is what allows ANN to have universal functions for a broad range of application, so that there is no specific input that is needed to be utilized. However, this implies that a fully connected network is weaker than a special-purpose network tuned to the structure of the problem space [27]. CNN explicitly assumes that the input are images, which allows for certain properties to be encoded into the model architecture [25]. ANN requires more computational power and has more information loss in the training phase due to the high pixel values in images; CNN is more adept in detecting features in images because it can decrease the high pixel values to lower

computational power and reduce information loss. CNN also requires less data for training and is faster to train than ANN [26].

2.4.2 CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN is gaining momentum in crack detection because of its accuracy and the increase in demand for autonomous methods due to the abundance of concrete structures in need of repairs. CNN is primarily used for image classification and object recognition because it explicitly uses images as input source. So, it has become a powerful image-based processing technique. CNN is a type of artificial intelligence that uses machine vision to focus on the most critical parts of the image in relation to the desired application [37]. So, the design of the CNN model is geared towards the specific desired function and application. In the case of crack detection, CNN can be manipulated to perform in the way the operator needs it to. For example, CNN can be used to predict crack measurements, predict crack depth, and classify the severity of damage, so according to the operator's problem space and goals.

In a previous study with CNN on crack detection, 1455 images of real concrete surfaces were used to train, validate, and test the CNN; 1250 of the images were cropped into a total of 60,000 smaller images of reduced pixel resolutions to build the training and validation set [29]. The remaining 205 images were used for testing. The training set is used for CNN to learn, and the validation set is used to fine tune the parameters to best fit the problem [30]. The parameter of the base learning rate of 0.01 was selected because it resulted in the highest validation accuracy of 99.06%, this base learning rate was used to test the remaining images. Along with CNN, this study also coupled exhaustive search with a sliding window to allow the cracks to be separated from the images as shown below.

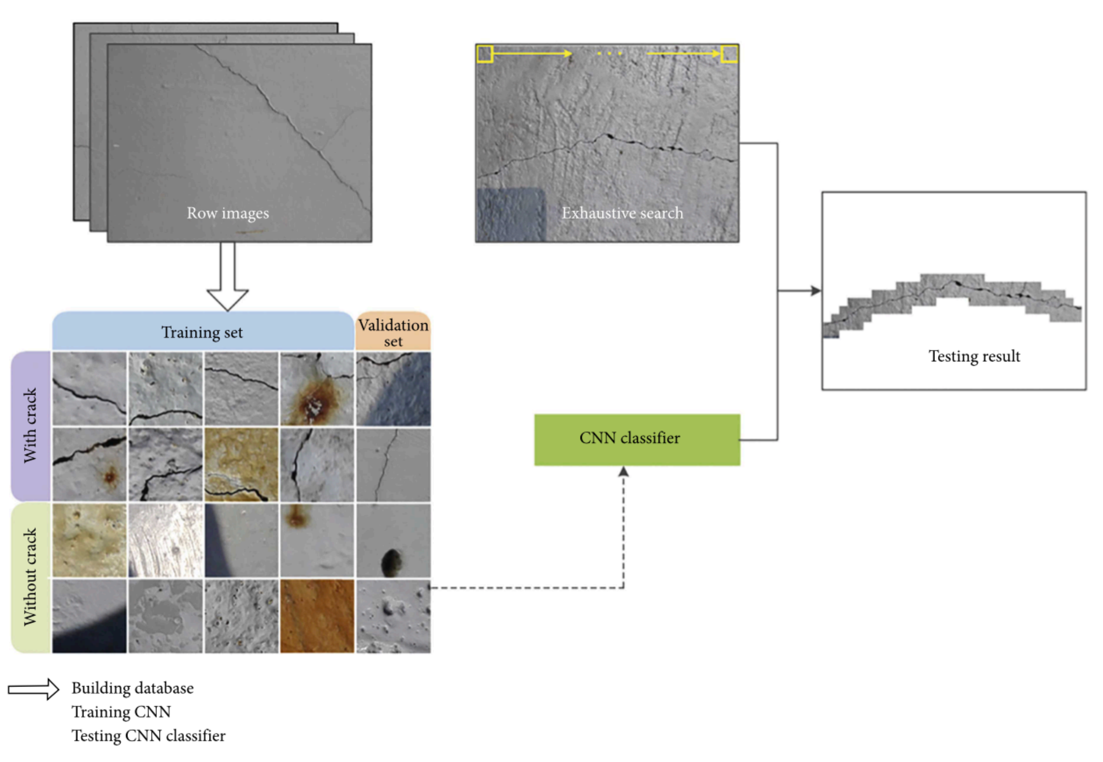


Figure 2.10 Separating Cracks From Images Using Exhaustive Search with a Sliding Window

After training, the CNN classifier can distinguish between images with cracks and images without cracks, then the exhaustive search allows for the crack to be extracted from the image.

The average testing accuracy for the test images was 99.09% [29]. The testing accuracy and validation accuracy is very close with a percent difference of 0.03% showing that the results are very precise and close together. These results demonstrate the suitability and reliability of CNN on crack detection. However, like all methods, there are obvious obstacles that need to be addressed while using CNN.

CNN needs a very large dataset to be able to train the model as accurately as possible. Having a small dataset will result in the model overfitting the data. This would

lead to exaggerated accuracy because not a wide enough range of images were available for use and only have biased accuracy towards the dataset used for training.

2.5 CONCLUSION

Advancements in the field of computer science allows for the continuous improvement of crack detection techniques. Civil engineers and researchers are leaving manual inspections of concrete cracks behind and have begun focusing on more complex computational techniques leading to the development of image processing methods. IPT results suggests that it is an effective method in identifying cracks and predicting their measurements, however, there are many challenges and obstacles that arise when trying to use IPT; noise from rough textures, shadows, and blemishes may increase the number of false positives and lower accuracy. To overcome the drawbacks of IPT and improve the performance of image-based crack detection, deep learning methods have been in development for crack detection due to their preprocessing ability to filter out noise to improve the quality of the image dataset. This produced reasonable predictions of length and width of concrete cracks, but there are not many studies on depth prediction of the crack.

CHAPTER 3

RESEARCH DESIGN

3.1 METHODOLOGY PERTAINING TO AIMS AND OBJECTIVES

To achieve the goals of this research stated in the objectives and scopes in section 1.2, this study was designed to meet each of the goals, as shown below:

1. Previous works have been completed on crack detection through manual inspections such as UPV, and AE. IPT is another popular method used for crack detection that overcomes the limitations in manual inspections such as labor intensity and lack of time-efficiency. These include edge detection and image binarization. These methods produce satisfactory results in crack detection, however, there are obvious obstacles that need to be overcome for IPT to become practical. Many images have background noise that include surface texture or roughness, shadows, blemishes, or holes; these noise factors may increase the number of false positive results. So, machine learning methods have been developed to create an autonomous method of crack detection.
2. In this study, CNN, XGBoost, and Random Forest of regression are utilized to evaluate images to detect cracks and predict depths after training. After the training and validation period, very minimal amount of human intervention is required to test images for cracks, so the possibility of autonomous inspections is very real.
3. The binary classification step with CNN in this study is designed to classify images whether a crack is present. This is attained by feeding the model positive images

(i.e. images with cracks) and negative images (i.e. images without cracks) to train the model to extract the features to help distinguish between cracks and no cracks in the test dataset

4. To classify the cracks into different damage zones to indicate the severity of the crack, cracks with known depth values were measured and labeled to be used in the dataset to train the multiclass CNN model to predict damage zones for each crack.
5. To predict the depths of cracks in the testing dataset, XGBoost and Randon Forest of regression models were used.

3.2 RESEARCH SIGNIFICANCE AND MOTIVATION

Concrete health monitoring is crucial in preserving the wellbeing of bridges, buildings, beams, and other important social infrastructures. To save money and preserve the public's safety, detecting cracks in its early stages is key to preventing further damages and injuries. Therefore, it is important to prioritize the more severe cracks first. Not many studies have been done on crack depth prediction, but it can become a great indicator for the intensity of the crack. For this research, the classification of severity is pursued based on prediction of crack depths.

Many traditional methods are becoming outdated and less efficient in detecting damages and cracks. This is due to the current state of concrete structures being below average, so the need of a quicker method of crack detection is increasing along with the need of repairs. In addition, there is a decrease in number of skilled inspectors. So, there is an urgent need of a more efficient and productive method of crack detection. This study proposes the possibility of autonomous crack detection utilizing deep learning to increase time-efficiency and reduce labor.

CHAPTER 4

EXPERIMENTAL SET UP

4.1 INTRODUCTION

Many concrete infrastructures such as bridges are deteriorating due to many factors such as age, weather conditions, and weight loading. Many cracks are not detected until the infrastructure needs severe repairs; however, detecting damages and cracks are imperative in preventing further damages and preserving the safety of the public. Due to the current state of infrastructure being below average in the United States, the need for effective crack detection techniques have increased. Many studies have been done to utilize machine ` techniques in identifying cracks and predict to enable crack detection to be autonomous to increase time-efficiency and reduce labor. Machine learning methods such as CNN and extreme gradient boosting (XGBoost) make it possible to detect cracks through image-based techniques and predict the depth of the crack to determine the severity of the crack.

With the increase in the need of structural repairs, classifying cracks based on severity has become an important aspect in crack detection. Many cracks range from hairline cracks to more severe cracks. Determining the severity of cracks allow civil engineers to prioritize severe cracks that are in dire need of attention and repairs. Prioritizing severe cracks lowers the risk of total or partial structural failure. Less severe cracks pose less safety risks, therefore, it is important to classify the cracks based on

severity for the safety of the public. In this study, there are three major steps used to assess the severity of the cracks demonstrated below.

1. One of the first steps to structural monitoring is to detect surface cracks, because it is one of the biggest indicators of further damage within the structure that cannot be detected by the naked eye. Autonomous binary classification allows for cracks to be identified without the need of human intervention. This allows for increased time-efficiency and reduction of labor because many traditional crack detection methods require inspectors to physically be present to test and analyze damages. CNN allows for binary classification of cracks through feature extraction to identify the presence of a crack in an image. It is trained through images containing cracks and images without cracks. After training the model, people with little knowledge of machine learning are still able to utilize the model which makes this method easy to be used in the field.
2. After identification of cracks, multiclass CNN is a method that allows for classification of different levels of damage zones to be assigned to each crack. This type of CNN is different from binary classification because it allows for cracks to be classified into three or more classes of output, whereas binary classification is based on one or two classes of output. This provides an insight on the severity of the cracks. Measured values of specimen crack depths are manually recorded and used in the multiclass CNN model to train it to recognize the severity of the crack to place it in its respective damage zone.
3. Knowing the depth levels of cracks is also an important aspect of crack detection. Many previous studies have been done on IPT and machine learning to detect

cracks and predict lengths and widths, however, not many studies have been done on depth prediction of cracks. This study utilizes XGBoost, an implementation of gradient boosting, to improve the strength of machine learning models and lower computational speed; XGBoost has been proven to outperform other data mining models such as decision tree regression and support vector machines [38]. Another method of crack detection that is investigated in this study is Random Forest tree. The two regression models will be compared to find the more accurate and precise model. Determining the more reliable method of crack detection can become beneficial to civil engineers and concrete inspectors.

4.2 SPECIMEN COLLECTION

The main specimen that was used in this project was a concrete slab. This slab was obtained in the civil engineering structural lab in 300 Main Building at the University of South Carolina located in Columbia, South Carolina. There are multiple cracks of varying lengths and depths running vertically along the surface of the slab (60in. x 8.25in. x 168in.). The cracks spanned across 1/3 of the width of the slab, so the lengths of the cracks are approximately 20 inches long. The cracks were measured at the deepest part; however, the whole crack does not have a uniform depth and varies throughout the length of the crack. Images and videos of cracks were captured using an iPhone 11 camera. The images have an original pixel resolution of 3024x4032, and a total of 15 raw images were taken. A total of four videos of each individual primary crack were recorded; the videos ranged from 13 seconds to 23 seconds. The specimens were panoramically recorded to include a wide range of angles to portray in the dataset when generating more images from the videos. This was done by investigating the frames of the videos to extract optimal images to be

used for the training process. The pictures and videos were taken during the day with sufficient lighting conditions and taken from approximately one to one and a half feet away from the surface of the cracks. For preprocessing, these images were cropped and expanded to generate a larger database of images. The cracks chosen on the slab had varying depths to account for a wider range in the dataset of images. This was done in hopes of increasing accuracy for a better representation of the model for the cracks. Four primary cracks were chosen and labeled 1 to 4. On each of the primary cracks, four secondary cracks (i.e., smaller cracks that stems or propagates from the primary crack) are chosen and labeled with decimals such that the secondary cracks on primary crack 1 is labeled 1.1 to 1.4. This is demonstrated in the figures below.

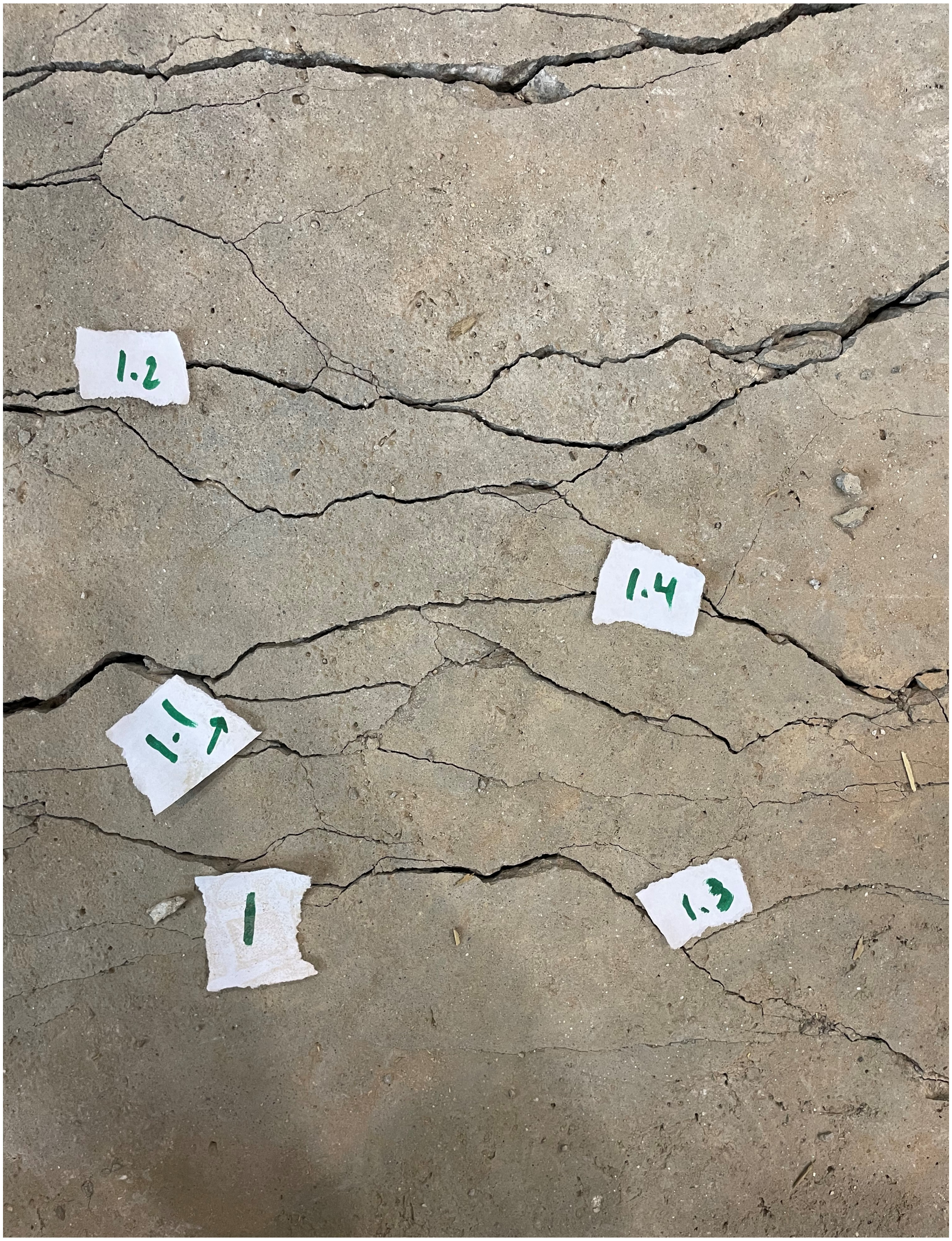


Figure 4.1 Visual Representation of Primary Crack Specimen 1 with its Secondary Cracks

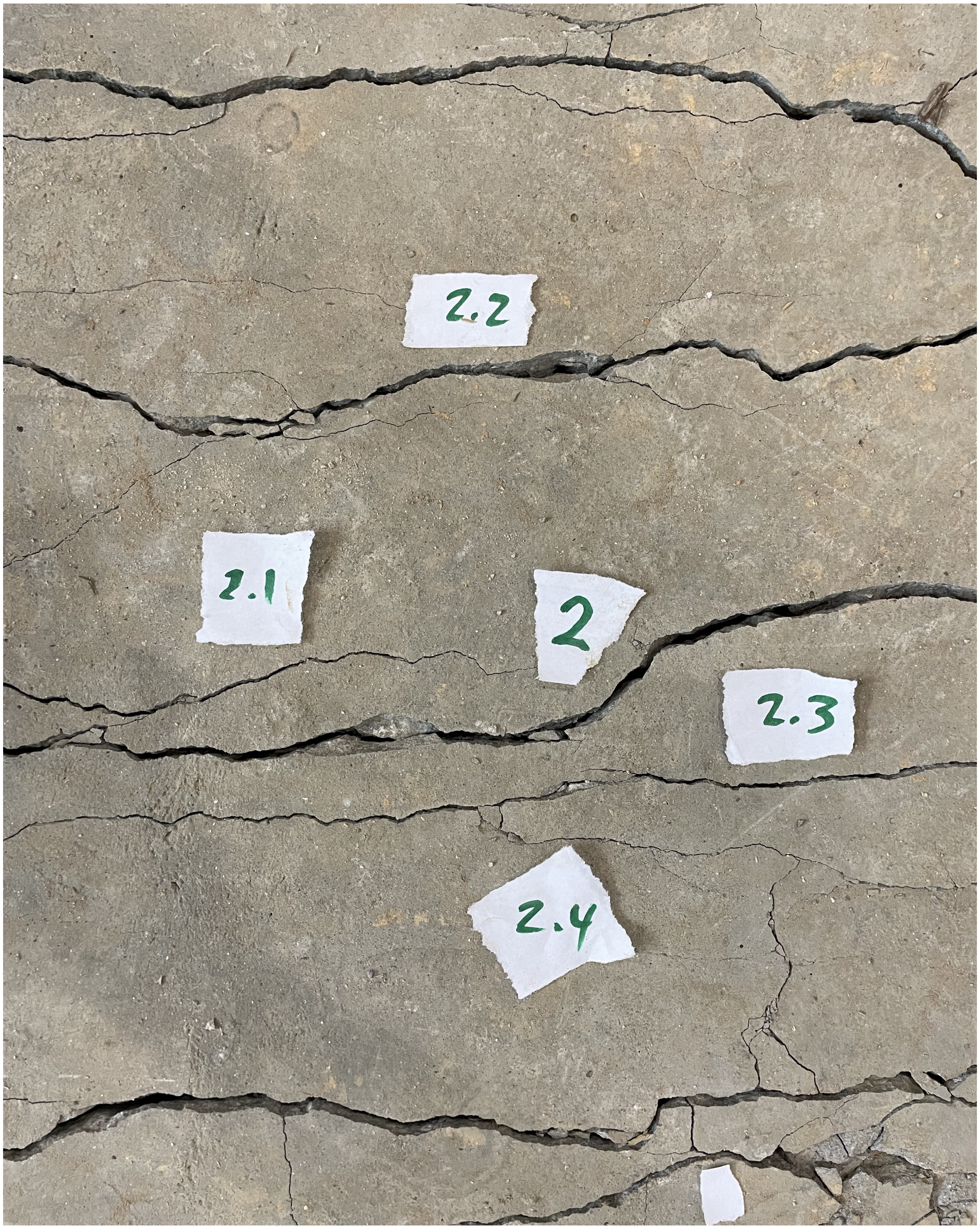


Figure 4.2 Visual Representation of Primary Crack Specimen 2 with its Secondary Cracks

Figure 4.3 Visual Representation of Primary Crack Specimen 3 with its Secondary Cracks

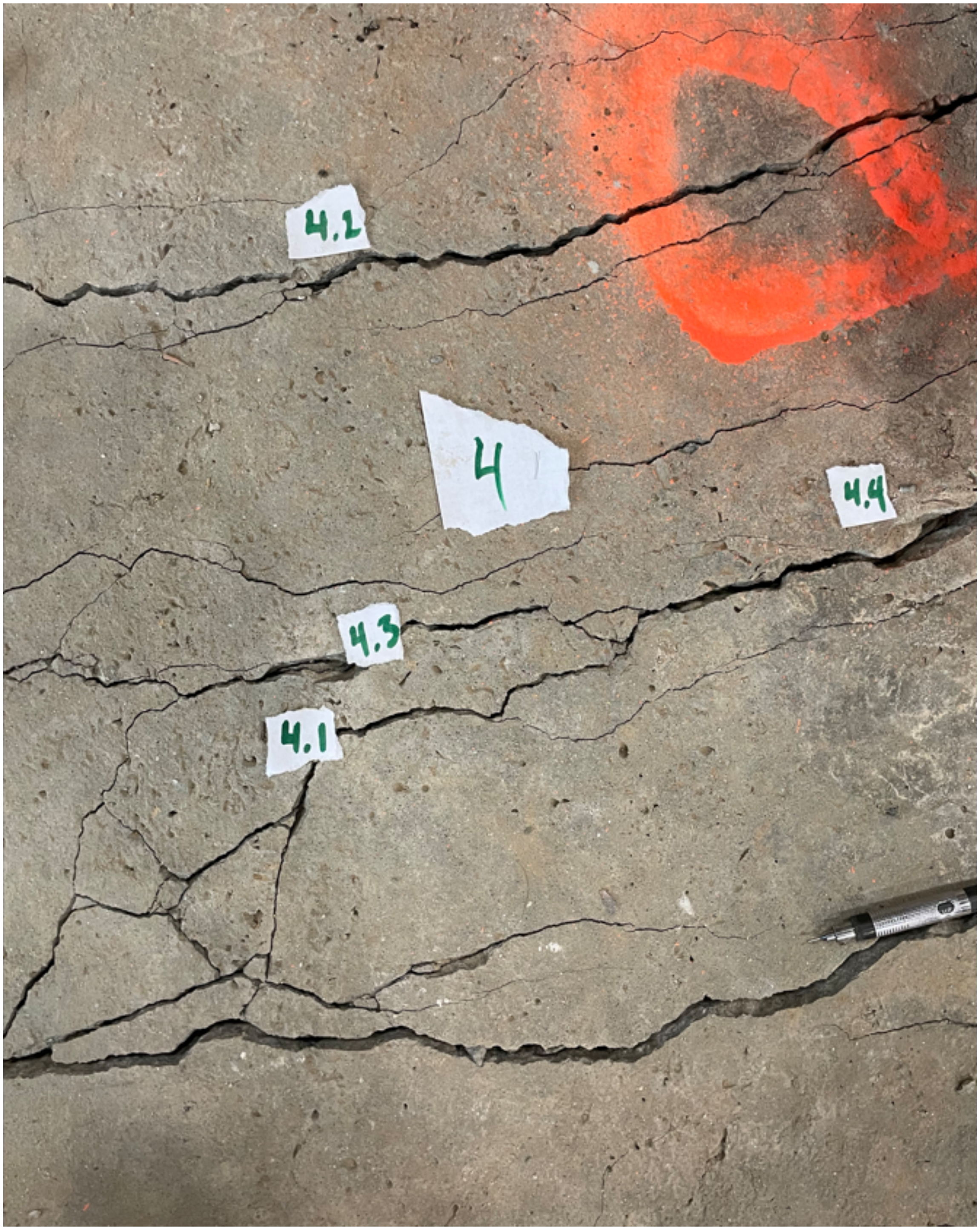


Figure 4.4 Visual Representation of Primary Crack Specimen 4 with its Secondary Cracks

Each primary and secondary crack is manually annotated to record the known depths of the crack at a certain point. This was done by taking a slip of paper and inserting it into the crack until it reached the bottom and marked it; then the length of the submerged part of the paper was measured with a ruler in centimeters.

4.3 MANUAL IMAGE PREPROCESSING FOR THE DATASET

The 15 raw images were cropped manually into smaller images to extract individual cracks from the large group of cracks shown in figure 4.5. So, the labeled images in section 4.2, were used as reference to compare to the original images to identify the cracks without the slips of labeling because it may affect the result and accuracy of the models if the slips of paper were left in the training process. So, four out of the 15 images were used as references. All the photos were used to crop the individual cracks into new images; this is shown in the figure below.

Figure 4.5 Original Image of Primary Cracks without Secondary Labeling Used for Manual Preprocessing to Produce Cropped images of Individual Cracks

From the raw image above, the individual primary and secondary cracks were cropped into smaller pictures with the single crack itself. This is shown in the figure below.

Figure 4.6 One Individual Cropped Image of Crack 1.1

This process was repeated for all the raw images and images from the videos to create a subset of images for each crack. This is shown in the figure below.

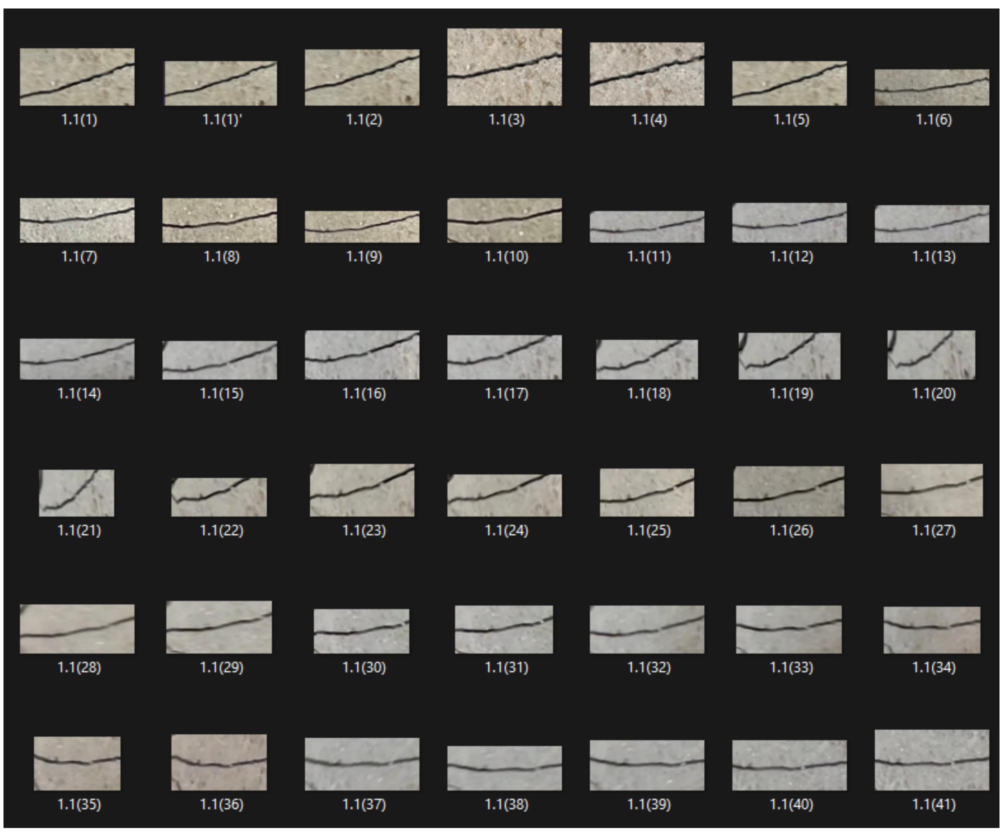


Figure 4.7 Subset of Images of Crack 1.1 Used for Training

There are a total of 20 folders of images; four of the folders are major cracks, and the other 16 are of secondary cracks. Each of the folders contain a range of 20-41 images. The measured depths of the 20 cracks are shown in the table below.

Table 4.1 Measurements of Crack Depths of Test Specimens

Crack Reference	Depth(cm)
1	1.1
1.1	0.1
1.2	2.5
1.3	0.2
1.4	1.8
2	3.1
2.1	1.5
2.2	3.3
2.3	2
2.4	0.1
3	1
3.1	1.2
3.2	0.1
3.3	0.5
3.4	2.1
4	0.4
4.1	1.4
4.2	2.3
4.3	1.4
4.4	2.5

From the crack depths, the damage zones were assigned so that three subsets of damage zones 1,2, and 3 were created, with 3 being the most severe. Damage zone 1 is classified as cracks with depths less than or equal 1 cm, damage zone 2 is classified as cracks with depths less than or equal 2 cm, and damage zone 3 involves the cracks that are deeper than 2 cm. This is depicted in the table below.

Table 3.2 Classification of Damage Zone Based on Depth

Zone Classification	Depth Range
Damage Zone 1	<= 1.0cm
Damage Zone 2	> 1.0cm & <= 2.0cm
Damage Zone 3	> 2.0cm

The color coding shown above, in table 3.1 and 3.2 demonstrate the damage zone for each crack. Red indicates damage zone 3, yellow demonstrates damage zone two, and green shows damage zone 1. The preprocessed dataset of images is only used in the multiclassification and depth prediction steps, whereas for binary classification, a public dataset was obtained from Kaggle.

4.4 BINARY CLASSIFICATION

For identification of cracks, a binary classification CNN model was trained to recognize the features of images with and without cracks. To train the model, an online public dataset of concrete with and without cracks was used; these images were captured in METU campus buildings and retrieved from Kaggle [39]. Kaggle is an online open-source community by Google which allows users to find public data sets and machine learning models can be built in a web-based platform. There are a total of 40,000 images of 227x227 pixel resolution with RGB channels. These images were split into two sets of positive and negative images each with 20,000 images. Examples of the negative and positive images are demonstrated below.

Figure 4.8 Representation of Negative Images of Kaggle

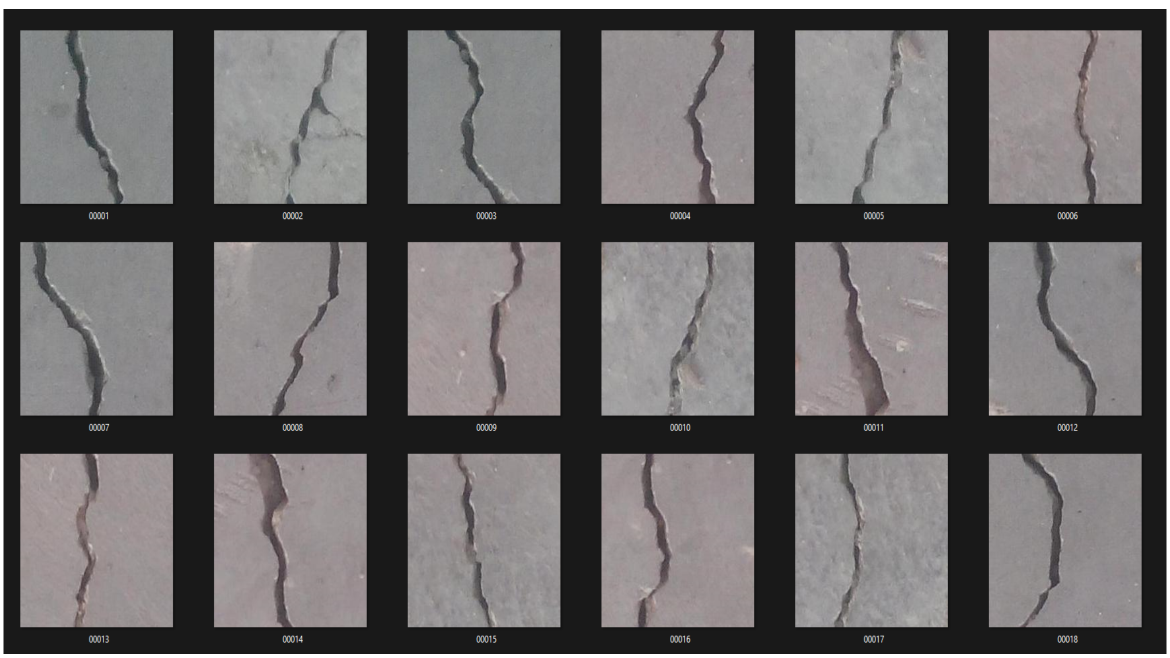


Figure 4.9 Representation of Positive Images of Kaggle

In the model, 22,400 images were used to train the CNN model with the crack versus non-crack ratio being 1 to 1. To validate the model, 5,600 images were used and to test the model after training, 12,000 were used. The training images are fed to the model, and the model learns and extracts the respective features for the positive and negative images.

Keras and TensorFlow were used in building machine learning models. Keras is a neural network Application Programing Interface (API) that is integrated with TensorFlow to develop and build the CNN in this study. Keras was selected because the framework is easy to use; it also gives clear feedback when there is a user error, so it makes it easy to learn as well [40]. The beginning portion of the code in training CNN model is shown below.

```
inputs = tf.keras.Input(shape=(120, 120, 3))
x = tf.keras.layers.Conv2D(filters=16, kernel_size=(3, 3), activation='relu')(inputs)
x = tf.keras.layers.MaxPool2D(pool_size=(2, 2))(x)
x = tf.keras.layers.Conv2D(filters=32, kernel_size=(3, 3), activation='relu')(x)
x = tf.keras.layers.MaxPool2D(pool_size=(2, 2))(x)
x = tf.keras.layers.GlobalAveragePooling2D()(x)
outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)

model = tf.keras.Model(inputs=inputs, outputs=outputs)

model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)

print(model.summary())
```

Figure 4.10 CNN Model for Binary Classification

In the model, the shape is 120x120x3, the three representing the RGB color mode, and the 120 is the target size of images. The kernel size that was selected is 3x3, a small kernel size was chosen to limit the number of unrelated features that are possible to be

filtered. A kernel is defined as the size of the window that slides across the image. The filter size for the first convolutional layer is 16 and 32 for the second layer. In the first layer, 16 lower-level features are extracted. In the second layer, 32 higher level features were extracted from the previous 16 features. Lower-level features include edges or lines in the image, whereas higher level features are the crack patterns, shapes, and sizes. A max pooling layer was used to calculate the maximum element from the region of the feature map under the filter; the output of the max pulling layer produces a feature map of the most prominent features of the previous feature map. A feature map is the results of the extracted features using the filter on the input images. A global average pooling (GAP) layer was also used, and it computes the average value of all elements in the feature map [41].

The Adam optimizer is one of the two arguments needed to compile the Keras model. Adam stands for Adaptive Moment Optimization, and it was chosen for its adaptability to smaller datasets, faster computational time, and fewer parameter needs for tuning [40, 42]. The final activation function used was sigmoid, and it is used for binary classification. Binary cross entropy was used for the loss function which is specifically tuned for binary classification; the loss indicates how close or far the predicted value is compared to the actual value with 0 being perfect and 1 being the worse, so the smaller the loss, the better the model [43]. After the CNN model is compiled, the model is fitted on the training set. A summary model of the CNN is demonstrated below.

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 120, 120, 3)]     0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 118, 118, 16)      448
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 59, 59, 16)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 57, 57, 32)        4640
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 28, 28, 32)        0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 33
=================================================================
Total params: 5,121
Trainable params: 5,121
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 4.11 CNN Model Summary of Binary Classification

4.5 MULTI CLASS CLASSIFICATION

A CNN model can be trained to recognize severity of cracks by classifying them into damage zones. The manual preprocessing required to generate the data, as described in section 4.3, is used in this step. After preprocessing and generating the dataset of images, each crack is sorted into a subset according to its respective damage zone for the training and testing dataset. For the training set, a total of 370 images were used; damage zone 1 has 124 images, damage zone 2 has 121 images, and damage zone 3 has 125 images. For the testing dataset, there are a total of 165 images used for evaluation of the model; damage zone 1 has 53 images, damage zone 2 has 54 images, and damage zone 3 has 58 images.

The known crack values in Table 4.1 are used to train the model to recognize the features to help predict depth of each crack to be used to gauge the severity of damages. Keras and TensorFlow were also used for this step of the study.

The main difference between binary classification and multiclass classification is that multiclass has three categories of output possibilities instead of two. The code below demonstrates the training of the CNN model.

```
cnn = tf.keras.models.Sequential()
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu', input_shape=[64, 64, 3]))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Flatten())
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))
cnn.add(tf.keras.layers.Dense(units=3, activation='softmax'))

cnn.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])

history = cnn.fit(training_set, validation_data=test_set, epochs=30)
print(cnn.summary())
```

Figure 4.12 CNN Model for Multi-class Classification

For this model, the input shape is 64x64x3. The filter sizes of both layers are 32 and the kernel size is 3x3. To compile the CNN model, the Adam optimizer was used, and the loss function used was categorical cross entropy (CCE). Unlike binary cross entropy, CCE is designed specifically for multi-class classification. Two max pooling layers were used, which allows the most prevalent features to be highlighted in each patch of each feature map; this reduces the number of parameters in the model and focus on the essential features of an image to prevent overfitting. A flattening layer was used to flatten the input to make it linear in a single dimension to pass onto a dense layer. The softmax was used as the final activation function for categorical classification, and the unit was three to represent the three damage zone outputs. Thirty epochs was chosen as a hyperparameter,

so the algorithm will pass through the entire training dataset 30 times to learn the features. A summary of the model is shown below.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_26 (Conv2D)           (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d_26 (MaxPooling (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_27 (Conv2D)           (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_27 (MaxPooling (None, 14, 14, 32)        0
_________________________________________________________________
flatten_13 (Flatten)         (None, 6272)              0
_________________________________________________________________
dense_26 (Dense)             (None, 128)               802944
_________________________________________________________________
dense_27 (Dense)             (None, 3)                 387
=================================================================
Total params: 813,475
Trainable params: 813,475
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 4.13 Summary of CNN Model Of Multiclass Classification

4.6 PREDICTION OF DEPTH

To predict depths of cracks, two regression models were used and compared. For this test, the models were trained with known crack depths from section 4.3. The features were extracted from the multiclass CNN model and fed into the regression models to predict trends. The regression models are trained to understand the relationship between

the independent variables and the outcomes. The independent variables in this case are the features from the multiclass CNN, and the output are the predicted damage zones. The known depths are important in feeding the regression model so that the model can understand the relationship between the independent and dependent variable. The two popular regression models that are evaluated in this test are Random Forest and XGBoost.

4.6.1 RANDOM FOREST SET UP

Random Forest is an ensemble learning method for regression, so the model takes the mean and average predictions of multiple decision trees and combines it to form more accurate predictions than just a single model or tree. Random Forest is a bagging technique that reduces risk of overfitting because each tree is created from a subset of data and the final output is based on the majority ranking or average; this reduces variance of single prediction since it combines several prediction trees from different models. Not all features are considered while creating for each individual tree, therefore, each tree is different. Random Forest produces stable results as the average outcomes given by many trees are taken. The original set of data is split into individual subsets for each decision tree model. The predictions all models are taken and combined to produce an average or majority ranking prediction. The features are extracted from multiclass CNN and fed to train the Random Forest regression model. The feature extraction layer is demonstrated below.

```
from keras.models import Model
layer_name='my_dense'
intermediate_layer_model = Model(inputs=cnn.input,
                                 outputs=cnn.get_layer(layer_name).output)

intermediate_layer_model.summary()
```

Figure 4.14 Feature Extraction Layer for Random Forest

The code above shows how the features are extracted from multiclass CNN and then fed to the Random Forest regression model. The extracted features were used to train and test the regression model which can be shown in the code below.

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error as MAE
from sklearn.metrics import mean_squared_error as MSE
from sklearn.metrics import r2_score


regressor = RandomForestRegressor(n_estimators = 50, random_state = 0)
regressor.fit(X_train_regression, y_train_regression)
y_pred_rf = regressor.predict(X_test_regression)
```

Figure 4.15 Random Forest Regression Model

From the sklearn package, RandomForestRegressor class was imported to create an instance of it, and assign it to a variable. The parameter *n_estimators* creates 50 decision trees in the Random Forest model. The *.fit( )* function used for training the model and adjusting weights according to the data values to achieve higher accuracy. The *.predict( )* function is used after training to make predictions. Four metrics were produced in the results to measure the performance and success of the model; the code to produce the metrics is depicted below.

```python
# MAE Computation
mae = MAE(y_test_regression, y_pred_rf)
print("MAE for Random Forest: % f" %(mae))
# MES Computation
mse = MSE(y_test_regression, y_pred_rf)
print("MSE for Random Forest: % f" %(mse))
# RMSE Computation
rmse = np.sqrt(MSE(y_test_regression, y_pred_rf))
print("RMSE for Random Forest: % f" %(rmse))
r2_score = r2_score(y_test_regression, y_pred_rf)
print("r2_score for Random Forest: % f" %(r2_score))
```

Figure 4.16 Random Forest Metrics

The four metrics produced are mean absolute error (MAE), mean square error (MSE), root mean squared error (RMSE), and r^2 value.

4.6.2 XGBOOST SET UP

XGBoost is another type of ensemble technique that utilizes gradient boosting. The decision trees in XGBoost are created in a sequential order, and the weight of the variables or each ensemble member is not the same throughout, whereas the weights in bagging of Random Forest were the same. As more decision trees are added, the model focuses on the variables that produced errors and places a higher weight on those variables. The higher weight variables are focused on and are then fed again to a subsequent decision tree to fit and correct the prediction errors made in the previous model. Boosting is a technique that reduces the bias of the model in training and minimizes overall prediction error. The features are also extracted from multiclass CNN and fed to train the XGBoost regression model. This is shown in the code below.

```python
import xgboost as xgb

model = xgb.XGBRegressor()
model.fit(X_train_regression, y_train_regression)
y_pred_xgb = model.predict(X_test_regression)
```

Figure 4.17 XGBoost Regression Model

To instantiate an XGBoost regressor object by calling the *XGBRegressor()* class from XGBoost library. The *.fit ( )* and *.predict ( )* functions were also used for XGBoost. The same four metrics were produced for XGBoost to measure the effectiveness of the model. The code to produce the metrics is displayed below.

```python
# MAE Computation
mae = MAE(y_test_regression, y_pred_xgb)
print("MAE for XGBoost: % f" %(mae))
# MES Computation
mse = MSE(y_test_regression, y_pred_xgb)
print("MSE for XGBoost: % f" %(mse))
# RMSE Computation
rmse = np.sqrt(MSE(y_test_regression, y_pred_xgb))
print("RMSE for XGBoost: % f" %(rmse))
r2_score = r2_score(y_test_regression, y_pred_xgb)
print("r2_score for XGBoost: % f" %(r2_score))
```

Figure 4.18 XGBoost Metrics

The four metrics produced are mean absolute error (MAE), mean square error (MSE), root mean squared error (RMSE), and r^2 value.

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 INTRODUCTION

Three different types of tests were performed in this study. Machine learning methods suggest the possibility of an alternative method to current traditional methods due to its accuracy and precision. The accuracies for every test were above 85% which indicates that machine learning can be utilized in crack detection. The binary classification test used a CNN model to identify the presence of cracks, the accuracy of the results was analyzed through confusion matrix. The second test performed was the multiclass classification of CNN to sort cracks into separate damage zones to indicate the level of severity. The results were analyzed using a confusion matrix, classification reports, training and validation loss plots, and training and validation accuracy plots. For the last test, two different models were explored in depth perception, XGBoost and Random Forest. These results showed that XGBoost model has better accuracy through a higher r^2 value compared to Random Forest regression model.

5.2 BINARY CLASSIFICATION RESULTS

A confusion matrix was used to measure the performance of the model to show the accuracy and precision of the predictions. Confusion matrices show the actual values versus the predicted values in solving classification problems. The confusion matrix for binary classification is displayed in the figure below.

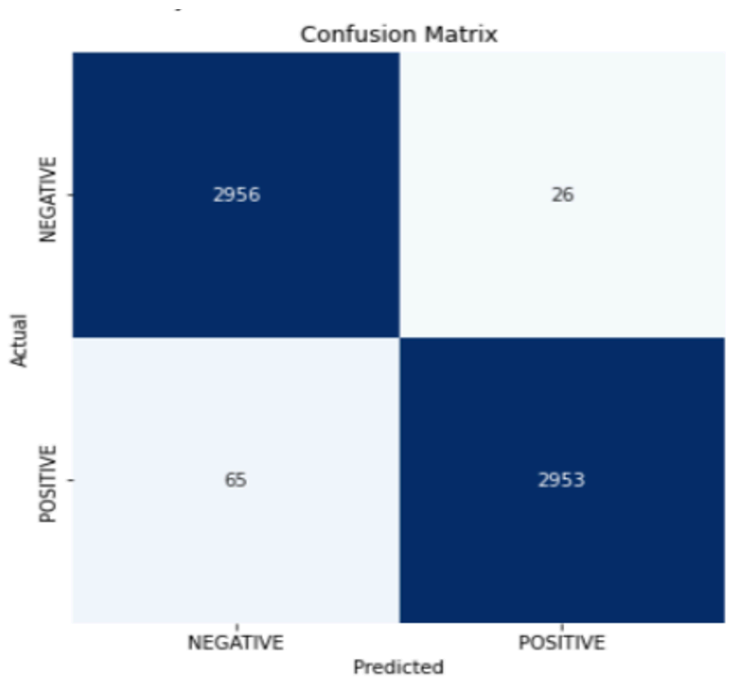


Figure 5.1 Confusion Matrix of Binary Classification CNN Results

A total of 6000 images were used as the test dataset. The model predicted 2953 images correctly as true positive, and it predicted 2956 images correctly as true negative. For the incorrect predictions, the model predicted 65 false negative images when, they were positive; the model also predicted 26 false positive images when they were negative. The false negative and false positive outcomes are significantly lower than the true negative and true positive values. A total of 5909 out of 6000 images were correctly predicted by the binary classification model and produced a test accuracy of 98.48% on crack detection.

A classification report is another evaluation metric used to measure the quality of predictions from the CNN model, and it is displayed below.

```
Classification Report:
----------------------
              precision    recall  f1-score   support

    NEGATIVE       0.98      0.99      0.98      2982
    POSITIVE       0.99      0.98      0.98      3018

    accuracy                           0.98      6000
   macro avg       0.98      0.98      0.98      6000
weighted avg       0.98      0.98      0.98      6000
```

Figure 5.2 Classification Report of Binary Classification CNN

The classification report produced precision, recall, and F1 scores for each class in the model. The figure below shows how each of the scores are calculated.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

$$Recall(TPR) = \frac{TruePositives}{TruePositives + FalseNegatives}$$

$$F_1 = \frac{2 * (precision * recall)}{(precision + recall)}$$

Figure 5.3 Equations to Calculate Precision, Recall, and F1Score

The precision is the ratio of correctly predicted positive observations to the total predicted positive observations, and recall is the ratio of correctly predicted positive observations to all the observations in the actual class. Recall is the measures the model's ability to correctly find all positive instances. The F1 score is the weighted average of precision and recall, so it considers both the false positives and false negatives. The precision and recall of both the negative and positive classes ranged from 98% to 99%, and

the F1 scores produced for both classes was 98%. The closer the value is to 1, the more precise and accurate the model is, therefore, the F1 score in this binary classification model indicates that the model is precise and accurate in detecting the presence of crack. The macro average is calculated by adding together the metrics of each class divided by the total number of classes, and the weighted average is calculated by taking the mean of all F1 scores per class while considering each class's support (i.e., occurrences in class). Since the classes are balanced and close to each other, the weighted average did not vary from the macro average. The training and validation loss plot was observed to measure how well the model fits the training and validation dataset, and this is demonstrated below. These scores demonstrate that the binary classification model produces reliable results in detecting the presence of cracks from images.

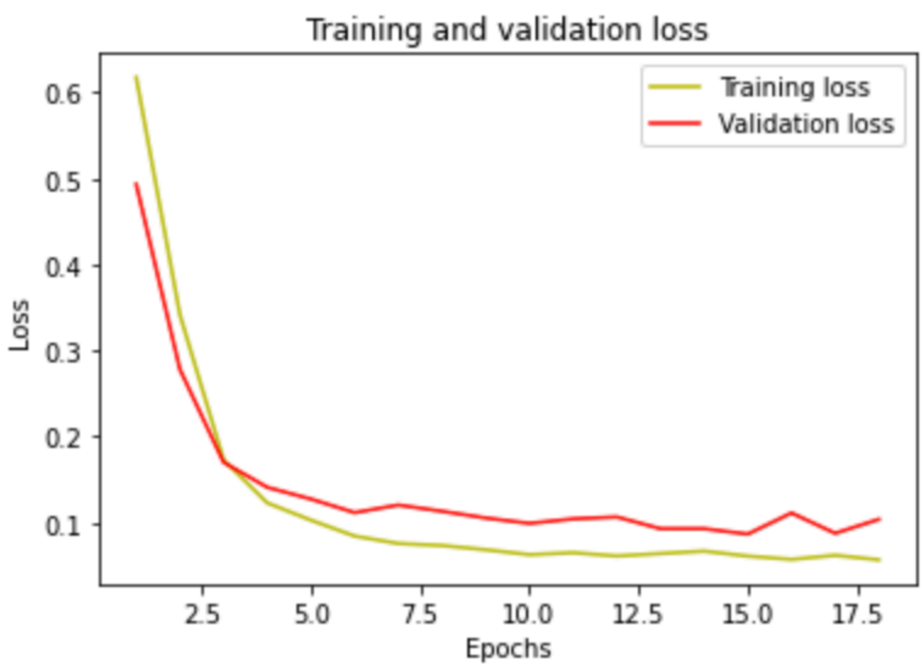


Figure 5.4 Relationship of Training and Validation Loss with Epochs in Binary Classification

Training and validation loss is the sum of errors made for each example in the training and validation datasets. The training loss shows how well the model is fitting the training dataset and the validation loss shows how well the model fits new data. The training loss is lower than the validation loss, which is expected because the validation data set is new, so it does not perform better than the training dataset. As the epochs increases, the training and validation loss both decreases, therefore the model is better fitting as the number of epochs increase. The training and validation loss plot above demonstrates that the model is well fitted to the new data in the validation dataset, therefore, binary classification of CNN produced fine results for identifying the presence of a crack in an image. The training and validation accuracy was also observed in the binary classification model, and this is displayed below.

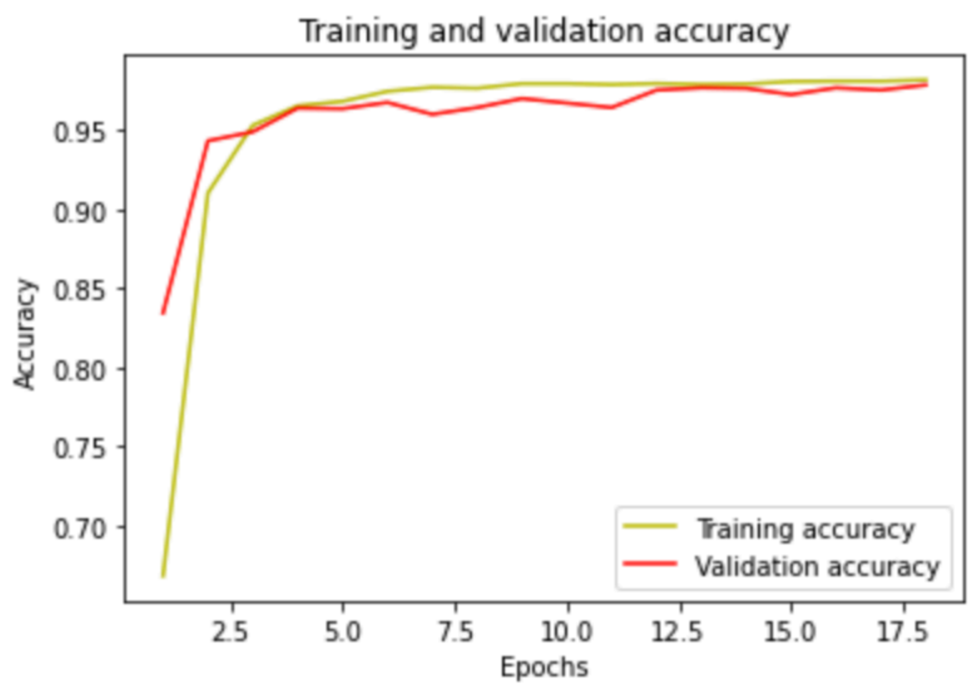


Figure 5.5 Relationship of Training and Validation Accuracy with Epochs in Binary Classification

Both the training and validation accuracy improves and increases as the number of epochs increase. The validation accuracy is slightly lower than the training accuracy and does not vary, therefore, it indicates that the model is accurate in classifying new data in the validation dataset from what it learned in the training dataset. Both accuracies are converging to 1 as the epochs increase, so it shows that the training dataset trained the model well and produced accurate predictions for crack detection in the validation dataset.

5.3 MULTICLASS CLASSIFICATION RESULTS

A confusion matrix was produced by the model to show the results of the multiclass classification of three damage zones. This is shown in the figure below.

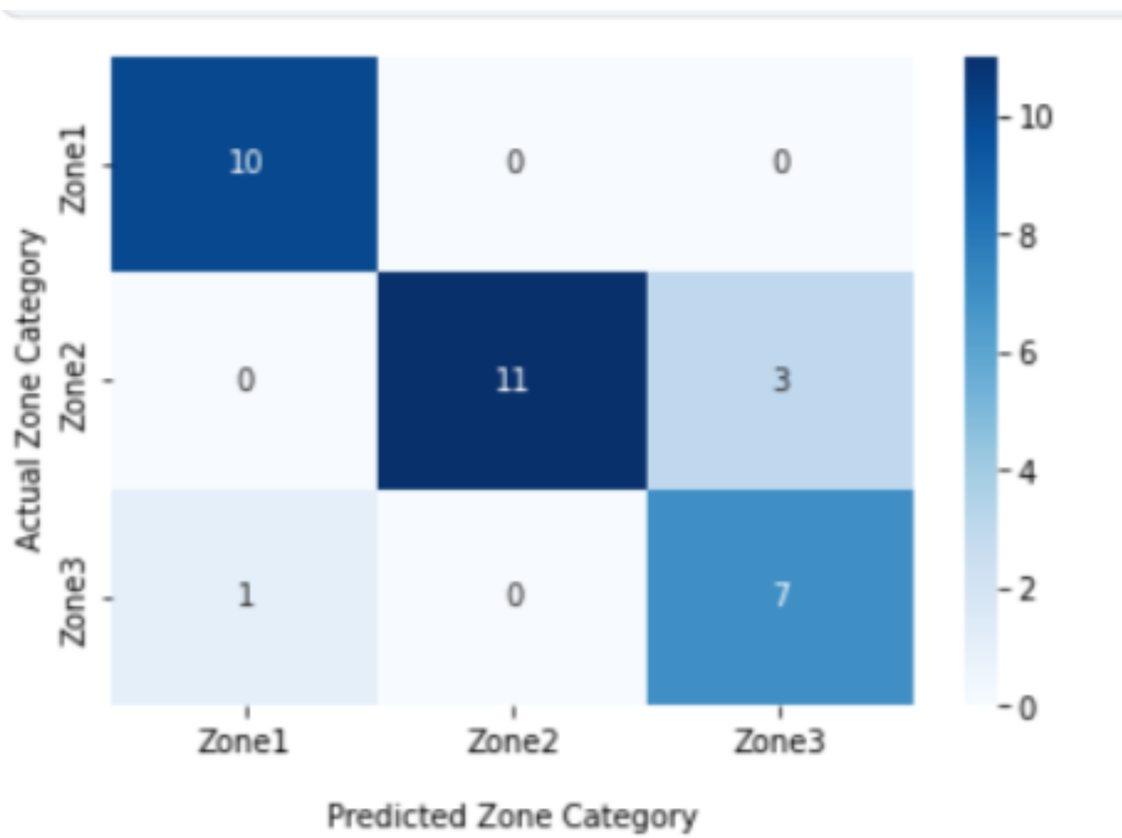


Figure 5.6 Confusion Matrix of Multiclass CNN Results

A total of 32 images were used to test the multiclass CNN model, 10 in damage zone 1, 14 in damage zone 2, and 8 in damage zone 3. The model predicted 28 out of 32 correctly which produced accuracy of 85%. One of the cracks in damage zone 3 was falsely predicted to be in damage zone 1, and three of the cracks in damage zone 2 was falsely predicted to be damage zone 3; so only 4 out of 32 images were falsely predicted by the model. This number is significantly lower than the number of correctly predicted damage zones, therefore, it demonstrates that the multiclass classification CNN model is accurate in predicting damage zones.

Classification Report:

----

	precision	recall	f1-score	support
zoneOne	0.91	1.00	0.95	10
zoneTwo	1.00	0.79	0.88	14
zoneThree	0.70	0.88	0.78	8
accuracy			0.88	32
macro avg	0.87	0.89	0.87	32
weighted avg	0.90	0.88	0.88	32

Figure 5.6 Classification Report of Multiclass Classification CNN

In this classification report of the multiclass classification CNN, the average or macro average of precision was 87% between the three damage zones. The precision is above 50%, so it can be considered that the predictions by the model are precise. Since there is class imbalance ranging from 8 occurrences in damage zone 3 to 14 occurrences in damage zone 2, the weighted average of precision is higher than the macro average due to more weight on the larger classes. In this case, damage zone three had the lowest precision and lowest occurrences, therefore, it has the least influence on the weighted average. The

average or macro average of recall was 89% and the weighted average of recall was 87%. The average or macro average of the F1 score is 88% and the weighted average of F1 score is 88%. With the accuracy, precision, recall, and F1 score all being above 85%, the multiclass classification CNN model can accurately predict damage zones from images.

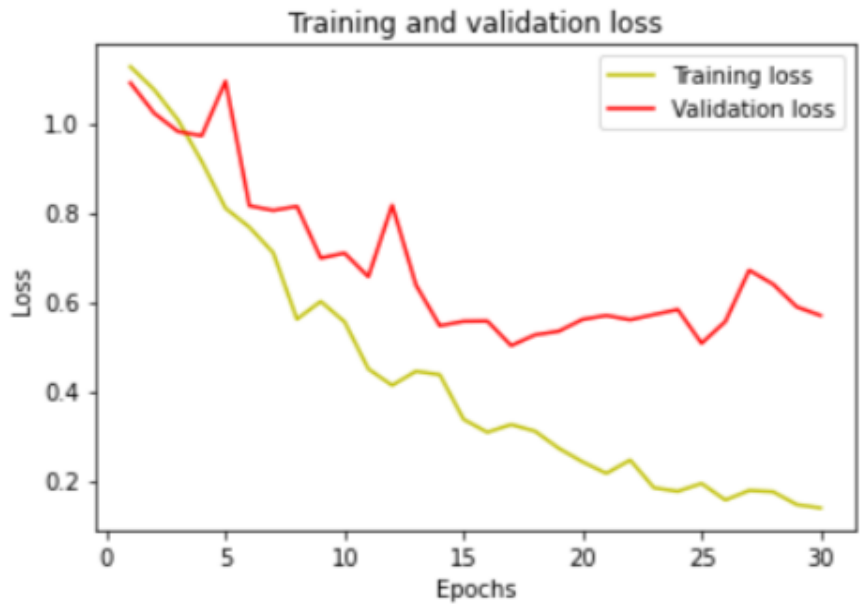


Figure 5.7 Relationship of Training and Validation Loss with Epochs in Multiclassification CNN

The training and validation loss both decrease as the number of epochs increases, however, the training and validation loss does not converge or flatten out to 0. The training loss is lower than the validation loss, this shows that the model performed well with the training dataset but not as well with new data in the validation dataset. However, it can still be concluded that the multiclass CNN model is able to predict damage zones accurately.

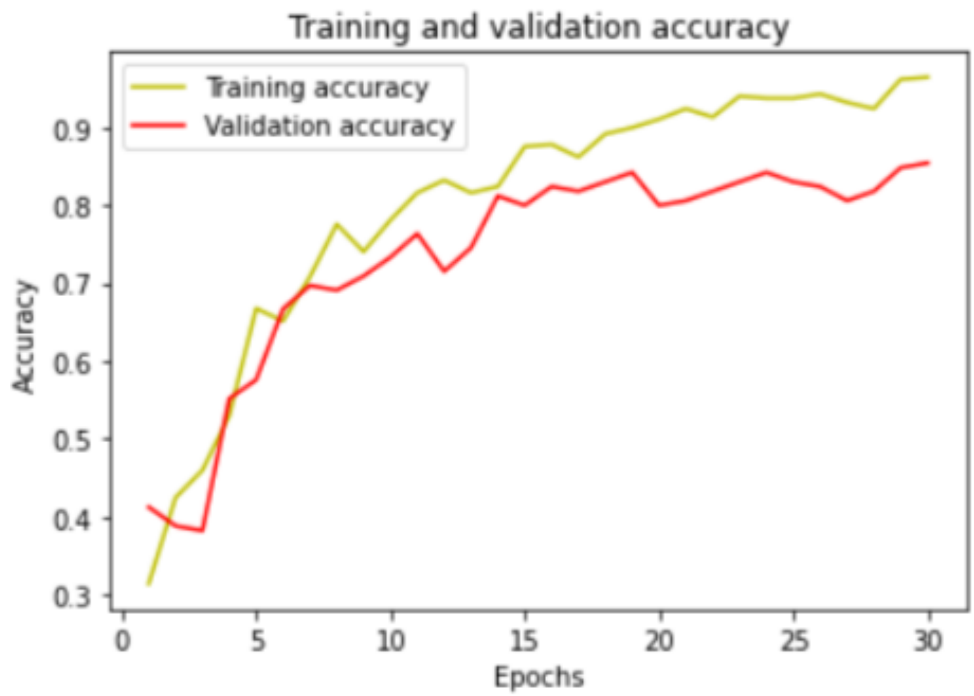


Figure 5.8 Relationship of Training and Validation Accuracy with Epochs in Multiclassification CNN

Both the training and validation accuracies increases as the number of epochs increase. This is because the more epochs and cycles the model progressed, the more the model has learned, thus increasing the accuracies. The training accuracy is slightly higher than the validation accuracy, however, they are still relatively close to each other. This indicates that the trained model fits the new data in the validation dataset and produced satisfactory results for accuracy. Therefore, this model is reliable in classifying damage zones of cracks.

5.4 REGRESSION MODELS RESULT

Two regression models were compared in predicting depth of cracks from a dataset of images with known crack depths. Both models produced satisfactory results with the lower having 0.88 r^2 value and the higher having 0.89 r^2 value. XGBoost outperforms

Random Forest, but only by 0.2% more, so both models are reliable in predicting crack depths from images. The metrics results for both models are shown in the figures below.

```
MAE for Random Forest:  0.237032
MSE for Random Forest:  0.142856
RMSE for Random Forest:  0.377963
r2_score for Random Forest:  0.888548
```

Figure 5.9 Results of Random Forest of Regression Model

```
MAE for XGBoost:  0.241883
MSE for XGBoost:  0.140905
RMSE for XGBoost:  0.375374
r2_score for XGBoost:  0.890070
```

Figure 5.10 Results of XGBoost of Regression Model

The four metrics that are used to measure the performance of the models are mean absolute error, mean squared error, root mean squared, and r^2. MAE represents the average of the absolute distance between the actual and predicted values in the dataset, and it measures the average of the residuals (i.e. prediction errors). Residuals are how far away the data points are in respect to the regression line. Therefore, the lower the MAE value is or closer to 0, the more accurate and less errors the model has. MSE represents the average of the squared distance between the original and predicted values in the dataset, and it measures the variance of the residuals. The lower the MSE, the more accurate the model is. RMSE represents the measurements of the average magnitude of errors and is concerned with the deviation from the actual value. A lower RMSE indicates the model with a better fit. The r^2 value represents how well the model accounts for the variance to see how many points fall onto the regression line, so the higher the r^2 value, the better the model is fitting the data and correlating. R^2 value is from 0% to 100%, with 100% meaning that every point

in the model and plot fits on the regression line; therefore, the higher the r^2 value, the better and more accurate the predictions are. In both models, the MSE, and RMSE were lower in XGBoost than Random Forest. The r^2 value was 88.8% for Random Forest and 89.0% for XGBoost. This demonstrates XGBoost as the model that produced the better and more accurate results, however, not by much, so both models are similarly accurate.

Two plots were created to show the relationship between the actual values and the predicted values to visualize each data point, these figures below demonstrate the plots for actual versus predicted values.

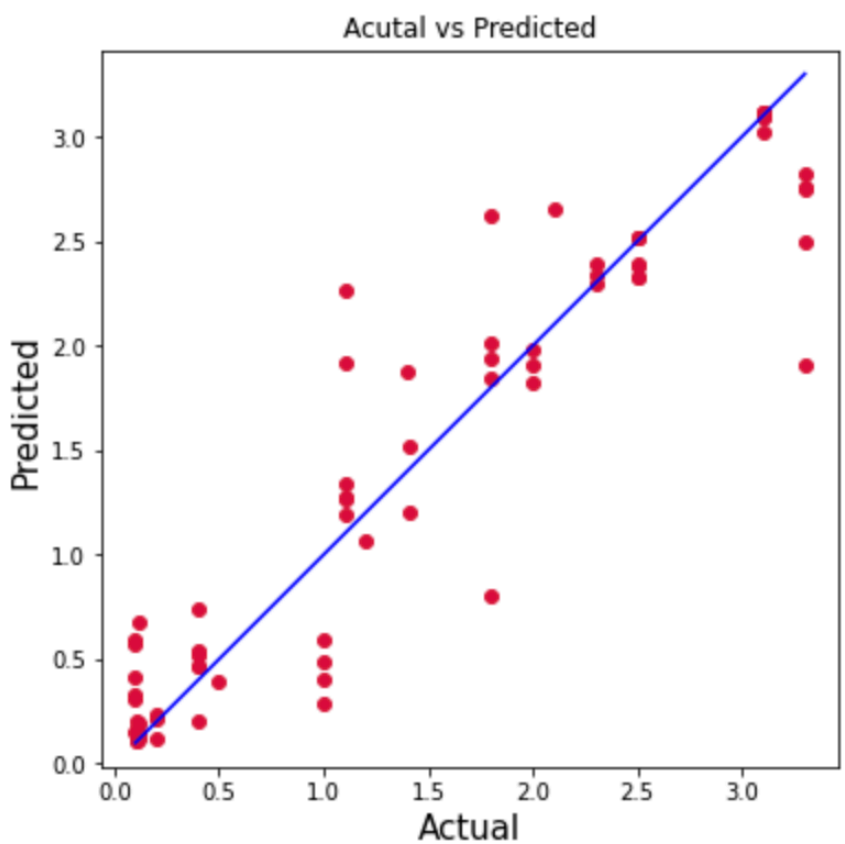


Figure 5.11 Actual vs. Predicted Values for Random Forest Prediction of Crack Depth

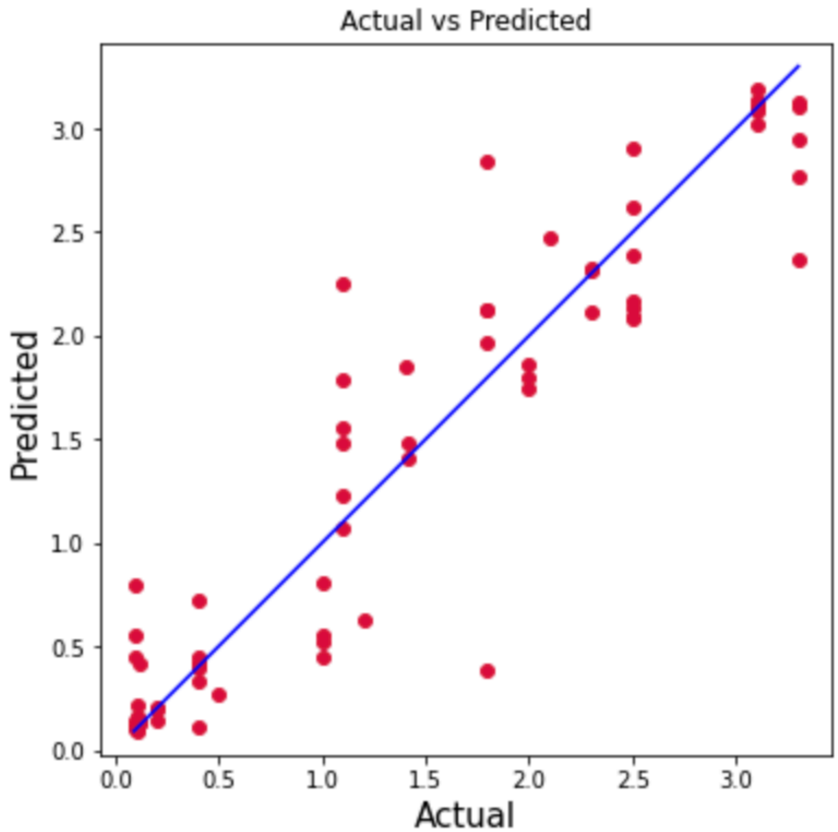


Figure 5.11 Actual vs. Predicted Values for XGBoost Prediction of Crack Depth

The figures above show the relationship between the predicted depth values from of the crack from the model with the actual known values. The closer the points are to the regression line, the more accurate the predictions are. Since XGBoost had the higher r^2 value, it indicates that more points in the plot of XGBoost are closer to the line of regression compared to plot of actual versus predicted in Random Forest.

CHAPTER 6

CONCLUSION

6.1 SUMMARY OF CRACK DETECTION

Crack detection has become a focus in the field of civil engineering due to the degradation of infrastructures in the United States. As time progresses, concrete structures become prone to damages from weight overload, cyclic loading and unloading, weather, and other factors which becomes a health hazard for the public. The number of skilled inspectors is slowly decreasing as the number of structures in need of repair and diagnoses are increasing rapidly, so the demand for an effective crack detection method is high. The current traditional methods are unable to keep up with this demand because traditional methods are both time consuming and labor intensive. In addition, untreated cracks degrade further which increases cost of repairs or may even require total replacement of a section in the concrete. So early intervention can help prevent further damages and restore structural integrity to ensure that the structure is safe to be used.

Crack detection can be used to monitor the overall health of structures. If there is a severe crack on the surface of the concrete, it may indicate that there may be internal damages that are not visible to the human eye. This is because water can enter the crack and overtime wear down the internal integrity of the concrete structure. Therefore, it is important to monitor the health of concrete structures through crack detection and categorizing the severity based on the depths of the crack to prioritize the cracks with the most damages.

6.2 STRENGTHS OF METHODOLOGIES

There are many strengths to the methodologies used in this research study. One strength is in the multiclass classification test and the regression models for predicting depths of cracks. The camera used was an iPhone camera, and the images were not of the highest quality as it would be with a high-quality image capturing device. So, the binary classification CNN model was able to train the model well even with lower quality images without it becoming an issue with the accuracy of the model.

Another strength to this study is through using ensemble techniques in predicting crack depths in the regression models. Both Random Forest and XGBoost are ensemble methods which take the predictions of multiple models then combine them to produce improved results. Random forest is a bagging technique which reduces the variance of a prediction model, whereas XGBoost is a boosting technique which reduces the bias and risk of overfitting in a prediction model. Ensemble methods produces more stable results compared to single models or a single decision tree model, so this improves the overall accuracy of the results for depth prediction in this study.

Using machine learning for crack detection and depth prediction significantly reduces the time consumption compared to the traditional crack detection methods. It lowers labor-intensity in addition to lowering costs of testing. After training and processing the machine learning models, very little human intervention is needed to analyze and produce results for damage assessment of concrete structures

6.3 LIMITATIONS AND FURTHER TESTING

Through this study, it can be suggested that regression models can be used for prediction of crack depths. The results of binary classification and multiclass classification

suggests that the model is effective in learning to detect cracks from a set of images, however, further work needs to be done in the field to evaluate the practicality. Although the results show promise to the field of civil engineering, there are many obstacles and limitations that are recommended for further investigations.

One recommendation of the binary classification methodology that should be explored in the future is the effect of the camera shot. The images in the online database provided images with the same camera distance, angle, and illumination. Most of the images taken in the field have varying camera shots, so it will be difficult to provide accurate results for field testing if the camera shot is not the same as the training images in the online database. To account for this, a wide variety of camera distance, angle, and illumination factors should be considered when training and validating the model; this would also promote accuracy for field testing. The accuracy result was 98.48% for this model, so it suggests this test method is effective, however, it cannot be concluded to be effective in the field because no field images were used to test the model in a real-world setting. So further testing of the model is required before introducing it as a reliable method in the field.

One limitation to the multiclass classification methodology is the limited number of images used to train the model. There was also no online dataset with images of cracks with known crack depth values, so further testing with more images is needed to test the practicality of the multiclass classification CNN model. CNN models require a large dataset to process and train to extract features from the images and learn from them. It produced a high accuracy of 85% with the limited images in the training dataset, however, with more images, the accuracy and precision of the model can be increased. When training

a machine learning model to recognize features, the quality and quantity of the images are directly proportional to the accuracy rate. Since the features were extracted from the multiclass CNN model, the limited training dataset also poses as an obstacle for training the regression models.

6.4 SUMMARY OF APPLICATIONS IN FIELD TESTING

This study showed that the use of machine learning methods in crack detection and depth prediction of crack are useful and can be applied in the field for further testing. The experiment produced satisfactory results that suggests its practicality and effectiveness in the field. The results produced for binary classification, multiclass classification, and depth prediction through regression models suggests that machine learning can be further investigated to be used in the field.

One application method of machine learning in crack detection is that it can be used alongside drone imaging. To further the reduce the need of human intervention, drones can be used to capture images without inspectors being physically onsite to capture the images of the cracks. Drones can capture images in a quicker manner and these images are then fed into the models after training. This would result a large quantity of data and increase the effectiveness of the crack detection. To make the model more robust, the drones can capture images with varying lighting conditions and these images can be utilized by the model. This would allow the process to become more autonomous and save time by eliminating the need of inspectors to travel to inspection sites. With the increase of structures in need of repairs and inspections, the methods in this research study proposes a reliable method to crack detection and depth prediction.

REFERENCES

[1] American Society of Civil Engineers (ASCE). (2017). Report card for America's infrastructure. Reston, Virginia.

[2] Kim, H., Ahn, E., Shin, M., & Sim, S.-H. (2019). Crack and Noncrack Classification from Concrete Surface Images Using Machine Learning. Structural Health Monitoring, 18(3), 725–738.

[3] Yang, C. H., Wen, M. C., Chen, Y. C., & Kang, S. C. (2015). An optimized unmanned aerial system for bridge inspection. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction* (Vol. 32, p. 1). IAARC Publications.

[4] Palevičius, P., Pal, M., Landauskas, M., Orinaitė, U., Timofejeva, I., & Ragulskis, M. (2022). Automatic Detection of Cracks on Concrete Surfaces in the Presence of Shadows. *Sensors*, *22*(10), 3662.

[5] Mohan, A., & Poobal, S. (2018). Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, *57*(2), 787-798.

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

[7] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)* (pp. 1-6). IEEE.

[8] Liu Y, Cho S., Spencer Jr, B.F., & Fan, J. (2014). Automated assessment of cracks on concrete surfaces using adaptive digital image processing. *Smart Structures and Systems*, 14(4): 719–741.

[9] Kim, H., Ahn, E., Cho, S., Shin, M., & Sim, S. H. (2017). Comparative analysis of image binarization methods for crack identification in concrete structures. *Cement and Concrete Research*, *99*, 53-61.

[10] Trtnik, G., Kavčič, F., & Turk, G. (2009). Prediction of concrete strength using ultrasonic pulse velocity and artificial neural networks. *Ultrasonics*, *49*(1), 53-60.

[11] Komlos, K., Popovics, S., Nürnbergerová, T., Babal, B., & Popovics, J. S. (1996). Ultrasonic pulse velocity test of concrete properties as specified in various standards. *Cement and Concrete Composites*, *18*(5), 357-364.

[12] Abdel-Qader, I., Abudayyeh, O., & Kelly, M. E. (2003). Analysis of edge-detection techniques for crack identification in bridges. *Journal of Computing in Civil Engineering*, *17*(4), 255-263.

[13] Mesquita, R. G., Silva, R. M., Mello, C. A., & Miranda, P. B. (2015). Parameter tuning for document image binarization using a racing algorithm. *Expert Systems with Applications*, *42*(5), 2593-2603.

[14] Tran, V. L., & Lin, H. Y. (2020). Extending and Matching a High Dynamic Range Image from a Single Image. *Sensors (Basel, Switzerland)*, *20*(14), 3950.

[15] Costin-Anton, B., Andrei-Iulian, D., & Dan-Cristian, C. (2009, August). Binarization for digitization projects using hybrid foreground-reconstruction. In *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing* (pp. 141-144). IEEE.

[16] Sauvola, Jaakko & Pietikäinen, Matti. (2000). Pietikainen, M.: Adaptive Document Image Binarization. Pattern Recognition 33, 225-236. Pattern Recognition. 33. 225-236. 10.1016/S0031-3203(99)00055-2.

[17] Jahanshahi MR, Masri SF, Padgett CW, et al. An innovative methodology for detection and quantification of cracks through incorporation of depth perception. Mach Vision Appl 2013; 24(2): 227–241.

[18] Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., & Fieguth, P. (2015). A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, *29*(2), 196-210.

[19] Wäldchen, J., & Mäder, P. (2018). Machine learning for image based species identification. *Methods in Ecology and Evolution*, *9*(11), 2216-2225.

[20] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521 (7553), 436-444. *Google Scholar Google Scholar Cross Ref Cross Ref.*

[21] Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, *4*(11), e00938.

[22] Mishra, M., & Srivastava, M. (2014, August). A view of artificial neural network. In *2014 international conference on advances in engineering & technology research (ICAETR-2014)* (pp. 1-3). IEEE.

[23] Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., ... & Chen, C. (2017). Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, *32*(10), 805-819.

[24] Kim, J. J., Kim, A. R., & Lee, S. W. (2020). Artificial neural network-based automated crack detection and analysis for the inspection of concrete structures. Applied Sciences, 10(22), 8105.

[25] Meng, M., Zhu, K., Chen, K., & Qu, H. (2021). A Modified Fully Convolutional Network for Crack Damage Identification Compared with Conventional Methods. *Modelling and Simulation in Engineering*, *2021*.

[26] Li, S., & Zhao, X. (2019). Image-based concrete crack detection using convolutional neural network and exhaustive search technique. *Advances in Civil Engineering*, *2019*.

[27] Gogul, I., & Kumar, V. S. (2017, March). Flower species recognition system using convolution neural networks and transfer learning. In *2017 fourth international conference on signal processing, communication and networking (ICSCN)*(pp. 1-6). IEEE.

[28] Washer, G., Fuchs, P., Graybeal, B. A., & Hartmann, J. L. (2004). Ultrasonic testing of reactive powder concrete. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, *51*(2), 193-201.

[29] Ali, L., Alnajjar, F., Jassmi, H. A., Gocho, M., Khan, W., & Serhani, M. A. (2021). Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures. *Sensors*, *21*(5), 1688.

[30] Li, S., & Zhao, X. (2018, March). Convolutional neural networks-based crack detection for real concrete surface. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018* (Vol. 10598, pp. 955-961). SPIE.

[31] Otsu, M. (1996). The history and development of acoustic emission in concrete engineering. Magazine of concrete research, 48(177), 321-330.

[32] Noorsuhada, M. N., Abdul Hakeem, Z., Soffian Noor, M. S., Noor Syafeekha, M. S., & Azmi, I. (2017, December). Correlation between average frequency and RA value (rise time/amplitude) for crack classification of reinforced concrete beam using acoustic emission technique. In *AIP Conference Proceedings* (Vol. 1901, No. 1, p. 050001). AIP Publishing LLC.

[33] Ohno, K., & Ohtsu, M. (2010). Crack classification in concrete based on acoustic emission. Construction and Building Materials, 24(12), 2339-2346.

[34] Polyzos, D., Papacharalampopoulos, A., Shiotani, T., & Aggelis, D. G. (2011). Dependence of AE parameters on the propagation distance. *J. Acoust. Emiss*, *29*, 57-67.

[35] Feng, X. T., Young, R. P., Reyes-Montes, J. M., Aydan, Ö., Ishida, T., Liu, J. P., & Liu, H. J. (2019). ISRM suggested method for in situ acoustic emission monitoring of the fracturing process in rock masses. Rock Mechanics and Rock Engineering, 52(5), 1395-1414.

[36] Yun, H.D.; Choi, W.C.; Seo, S.Y. Acoustic emission activities and damage evaluation of reinforced concrete beams strengthened with CFRP sheets. *NDT E Int*. 2010, *43*, 615–628.

[37] Yang, R., Singh, S. K., Tavakkoli, M., Amiri, N., Yang, Y., Karami, M. A., & Rai, R. (2020). CNN-LSTM deep learning architecture for computer vision-based modal frequency detection. *Mechanical Systems and signal processing*, *144*, 106885.

[38] Pan, B. (2018, February). Application of XGBoost algorithm in hourly PM2. 5 concentration prediction. In *IOP conference series: earth and environmental science* (Vol. 113, No. 1, p. 012127). IOP publishing.

[39] Özgenel, Ç. F. (2018). Concrete crack images for classification. *Mendeley Data*, *1*(1).

[40] Ramasubramanian, K., & Singh, A. (2019). Deep learning using keras and tensorflow. In *Machine Learning Using R* (pp. 667-688). Apress, Berkeley, CA.

[41] Hsiao, T. Y., Chang, Y. C., Chou, H. H., & Chiu, C. T. (2019). Filter-based deep-compression with global average pooling for convolutional networks. *Journal of Systems Architecture*, *95*, 9-18.

[42] Nhu, V. H., Hoang, N. D., Nguyen, H., Ngo, P. T. T., Bui, T. T., Hoa, P. V., ... & Bui, D. T. (2020). Effectiveness assessment of Keras based deep learning with different robust optimization algorithms for shallow landslide susceptibility mapping at tropical area. *Catena*, *188*, 104458.

[43] Ruby, U., & Yendapalli, V. (2020). Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, *9*(10).

[44] Lancaster, L. C. (2005). *Concrete vaulted construction in Imperial Rome: innovations in context*. Cambridge University Press.

[45] Sham, F. C., Chen, N., & Long, L. (2008). Surface crack detection by flash thermography on concrete surface. Insight-Non-Destructive Testing and Condition Monitoring, 50(5), 240-243.

[46] Lin, S., & Wang, Y. (2020). Crack-Depth estimation in concrete elements using ultrasonic shear- horizontal waves. Journal of Performance of Constructed Facilities, 34(4), 04020064.

[47] Sun, Y., Huang, P., Su, J., & Wang, T. (2018). Depth estimation of surface-opening crack in concrete beams using impact-echo and non-contact video-based methods. EURASIP Journal on Image and Video Processing, 2018(1), 1-10.

[48] Shen, S., Lu, X., Liu, L., & Zhang, C. (2016). Investigation of the influence of crack width on healing properties of asphalt binders at multi-scale levels. *Construction and Building Materials*, *126*, 197-205.