
Electronic Thesis and Dissertation Repository

8-20-2020 11:30 AM

Classification, Localization, and Quantification of Structural Damage in Concrete Structures using Convolutional Neural Networks

Majdi Flah, *The University of Western Ontario*

Supervisor: Moncef L. Nehdi, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Civil and Environmental Engineering

© Majdi Flah 2020

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>



Part of the [Civil Engineering Commons](#), [Computational Engineering Commons](#), [Construction Engineering and Management Commons](#), and the [Structural Engineering Commons](#)

Recommended Citation

Flah, Majdi, "Classification, Localization, and Quantification of Structural Damage in Concrete Structures using Convolutional Neural Networks" (2020). *Electronic Thesis and Dissertation Repository*. 7188. <https://ir.lib.uwo.ca/etd/7188>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlsadmin@uwo.ca.

Abstract

Applications of Machine Learning (ML) algorithms in Structural Health Monitoring (SHM) have recently become of great interest owing to their superior ability to detect damage in engineering structures. ML algorithms used in this domain are classified into two major subfields: vibration-based and image-based SHM. Traditional condition survey techniques based on visual inspection have been the most widely used for monitoring concrete structures in service. Inspectors visually evaluate defects based on experience and engineering judgment. However, this process is subjective, time-consuming, and hampered by difficult access to numerous parts of complex structures. Accordingly, the present study proposes a nearly automated inspection model based on image processing, signal processing, and deep learning for detecting defects and identifying damage locations in typically inaccessible areas of concrete structures. The work conducted in this thesis achieved excellent damage localization and classification performance and could offer a nearly automated inspection platform for the colossal backlog of ageing civil engineering structures.

Keywords

Machine Learning; Deep Learning; Convolutional Neural Network; Damage Detection; Damage Localization; Structural Health Monitoring; Image Processing; Signal Processing; Vibration; Concrete structures.

Summary for Lay Audience

Diagnosing damage in civil engineering structures and infrastructures has been getting increasing attention due to the very large portfolio of ageing civil assets and concerns related to its serviceability and safety. Until now, visual inspection has been the most used method to assess structural damage. However, in many cases, it is difficult and unsafe to access parts of such infrastructure (e.g., massive offshore bridge, large dam, tall building, etc.). At the same time, assessing damage can vary from one operator to the other depending on expertise and personal judgement. In this research, this subjectivity is mitigated using advanced statistical and probabilistic approaches such as artificial intelligence combined with image processing techniques to localize structural damage, quantify it, and even predict its type and degree of severity. This is done by implementing algorithms based on datasets of images for both damaged and intact structures. Then, depending on whether the structure is cracked or not, a quantification algorithm is developed to measure the width, length, and angle of orientation of cracks.

Nevertheless, in many cases, bridges, buildings and other structures have collapsed without presenting any warning signs, for instance via loss of the stiffness of key structural elements due to inner degradation that cannot be detected by visual inspection at the surface of the structure. For this reason, a global technique based on signal processing is needed. When a random excitation is applied to a building, and its acceleration signals are measured, then damage features from the signal can be automatically extracted. Accordingly, the position of damage can be determined. The contribution of this thesis in this area is part of larger effort to minimize and substitute to the subjective human operator in inspection and rehabilitation protocols. This study could, with further developmental work, optimize the service lifecycle, minimize maintenance costs, and mitigate failure risks for the lifetime of a civil infrastructure asset. Eventually, this research aims at making vital structures highly durable and long-lasting in Canada and worldwide. It might be very costly to erect new buildings and bridges, but we could give more life to the old ones at lower cost.

Co-Authorship Statement

Following the guidelines and regulations of the School of Graduate and Postdoctoral Studies (SGPS) at Western University, this thesis was organized according to the integrated-article format. Chapters 2, 3, and 4 of the thesis were submitted for peer-review and potential publication in reputed technical journals or conferences as listed below. The applicant, under the direct supervision of Prof. M.L. Nehdi, performed all the experimental research, numerical study, data analysis, and writing of the thesis and draft technical articles. Other co-authors aided in the final version of the publications or provided support in the experimental research and computational modeling.

Flah, M., Itzel, N., Chaabene, WB., and Nehdi, M. L. "Machine Learning Algorithms in Civil Structural Health Monitoring: A Systematic Review," submitted to *Archives of Computational Methods in Engineering*, 2020.

Flah, M., Suleiman, AR., and Nehdi, M. L. "Classification and Quantification of Cracks in Concrete Structures using Deep Learning Image-Based Technique", submitted to *Cement and Concrete Composites*, 2020.

Flah, M., and Nehdi, M. L. (2020)"Automated Crack Identification using Deep Learning Based Image Processing," National Conference on Infrastructure Management, Proceedings of the Annual Conference of the Canadian Society for Civil Engineering, Saskatoon, Canada.

Flah, M., Ragab, M., Lazhari, M., and Nehdi, M. L. "Localization and Classification of Structural Damage using Deep Learning Single-Channel Signal-Based Measurement," submitted to *Structural Health Monitoring*, 2020.

Dedication

To:

My Beloved Mother Jamila,

My Father AbdelJawad,

My Beloved Sister Hayfa,

And My Brother Aymen

Acknowledgments

First, I would like to thank my supervisor, Professor Moncef L. Nehdi, for believing in me and enabling me to be part of his research team. I am always thankful for his support, suggestions, motivation, and positive input, which allowed me to be even more successful and efficient.

I wish to acknowledge the financial support provided by the Canadian organization Mitacs within the program of Mitacs Globalink and Mitacs Graduate Fellowship (Intern ID # GLF580).

Special thanks to my co-authors Ahmed Ramadan Suleiman, Malek Lazhari, Mohamed Ragab, Afshin Marani, Wassim Ben Chaabene, and Itzel Nunez for their help, support, advice, cooperation, and valuable discussion during my work.

I would like to express my heartfelt gratitude to the Association of Arab Students (ASA) for offering moral encouragement, inspiration, and inclusion in the local community during my time here in London.

I appreciate all CEE Faculty and Staff for the enjoyable learning experience during my study time and thesis work.

Finally, I would like to thank my parents; brother and sister, and all my relatives and friends for all their patience, compassion and encouragement through the years. Without their unyielding motivation and assistance, this accomplishment would not have been feasible.

Table of Contents

| | |
|---|------|
| Abstract..... | ii |
| Summary for Lay Audience..... | iii |
| Co-Authorship Statement..... | iv |
| Dedication..... | v |
| Acknowledgments..... | vi |
| Table of Contents..... | vii |
| List of Tables..... | xi |
| List of Figures..... | xiii |
| List of Symbols..... | xvi |
| List of Abbreviations..... | xix |
| Chapter 1..... | 1 |
| 1 Introduction..... | 1 |
| 1.1 Background..... | 1 |
| 1.2 Research Gaps..... | 2 |
| 1.3 Research Need and Objectives..... | 4 |
| 1.4 Original Contributions..... | 5 |
| 1.5 Thesis Structure..... | 6 |
| 1.6 References..... | 7 |
| Chapter 2..... | 9 |
| 2 Machine Learning Algorithms in Civil Structural Health Monitoring: A Systematic Review..... | 9 |
| 2.1 Introduction..... | 9 |
| 2.2 Hierarchy of ML algorithms..... | 12 |
| 2.2.1 Input Configuration..... | 12 |
| 2.2.2 Algorithm Manipulation..... | 14 |
| 2.2.3 Output Manipulation..... | 16 |

| | | |
|----------------|---|----|
| 2.3 | SHM Subfields..... | 16 |
| 2.3.1 | Bridge Health Monitoring (BHM)..... | 16 |
| 2.3.2 | Building Health Monitoring (BUHM)..... | 17 |
| 2.3.3 | Dam Health Monitoring (DHM)..... | 17 |
| 2.3.4 | Wind Turbine Health Monitoring (WTHM)..... | 18 |
| 2.4 | DL and ML Applications in SHM | 19 |
| 2.4.1 | Artificial Neural Networks (ANNs)..... | 19 |
| 2.4.2 | Support Vector Machine (SVM)..... | 30 |
| 2.4.3 | Other Algorithms | 32 |
| 2.5 | Analysis and Discussion | 37 |
| 2.6 | Conclusions..... | 42 |
| 2.7 | Conflict of interest statement..... | 43 |
| 2.8 | References..... | 43 |
| Chapter 3..... | | 55 |
| 3 | Classification and Quantification of Cracks in Concrete Structures using 2D CNN Image-Based Techniques | 55 |
| 3.1 | Introduction..... | 55 |
| 3.1.1 | Research Significance..... | 58 |
| 3.2 | Data Manipulation | 59 |
| 3.2.1 | Data Preparation..... | 59 |
| 3.2.2 | Data Processing..... | 59 |
| 3.2.3 | Data Labelling..... | 60 |
| 3.3 | CNN Classification | 62 |
| 3.3.1 | Convolution layer..... | 62 |
| 3.3.2 | Pooling layer | 64 |
| 3.3.3 | Non-Linearity or Activation layers..... | 65 |
| 3.3.4 | Dropout layers..... | 65 |

| | | |
|-----------|---|-----|
| 3.3.5 | Batch Normalization | 66 |
| 3.3.6 | Softmax Layer..... | 67 |
| 3.3.7 | CNN Classifier Considerations..... | 68 |
| 3.4 | Image Segmentation..... | 69 |
| 3.4.1 | Image Improvement | 70 |
| 3.4.2 | Thresholding using Otsu’s Method..... | 71 |
| 3.4.3 | Noise Removal and Image Connection..... | 72 |
| 3.4.4 | Calculating Crack Dimensions | 73 |
| 3.4.5 | IPTs Considerations | 76 |
| 3.5 | Mode of Failure Prediction and Damage Severity Check | 77 |
| 3.5.1 | Prediction of Mode of Failure..... | 77 |
| 3.5.2 | Damage Severity Check..... | 80 |
| 3.6 | Results and Discussion | 82 |
| 3.7 | Conclusions..... | 92 |
| 3.8 | References..... | 93 |
| Chapter 4 | | 97 |
| 4 | Localization and Classification of Structural Damage using 1D CNN Single-Channel Signal-Based Measurement..... | 97 |
| 4.1 | Introduction..... | 97 |
| 4.1.1 | Research Significance | 101 |
| 4.2 | Methodology | 102 |
| 4.2.1 | 1D CNN Network Architecture | 102 |
| 4.2.2 | CNN Representation | 103 |
| 4.3 | Data Preparation..... | 109 |
| 4.3.1 | Data Description | 109 |
| 4.3.2 | Data Decomposition using the sliding window technique..... | 111 |
| 4.3.3 | Data Augmentation using SMOTE..... | 111 |

| | | |
|------------------|--|-----|
| 4.3.4 | Data Processing..... | 112 |
| 4.4 | Results and Discussions..... | 113 |
| 4.4.1 | Experimental Settings..... | 114 |
| 4.4.2 | Evaluation Metrics..... | 115 |
| 4.4.3 | Case Study 1..... | 118 |
| 4.4.4 | Case Study 2..... | 122 |
| 4.4.5 | Case Study 3..... | 124 |
| 4.4.6 | Effect of Noise..... | 126 |
| 4.5 | Conclusion and Future Work..... | 127 |
| 4.6 | References..... | 128 |
| Chapter 5 | | 132 |
| 5 | Conclusions, Recommendations, and Future Research..... | 132 |
| 5.1 | Conclusions..... | 132 |
| 5.2 | Recommendations and Future Research..... | 133 |
| Curriculum Vitae | | 135 |

List of Tables

| | |
|---|-----|
| Table 2.1: Summary of the Different NN Applications in SHM..... | 21 |
| Table 2.2: Summary of the Different BPNN Applications in SHM..... | 23 |
| Table 2.3: Summary of CNNs applications for SHM..... | 25 |
| Table 2.4: Summary of the Different SVM Applications in SHM..... | 31 |
| Table 2.5: Other ML Algorithms..... | 33 |
| Table 3.1: Dimensions of Layers and Operations..... | 63 |
| Table 3.2: Prediction of Mechanical Stress or Durability issues Causing Crack based on the Angle of Orientation..... | 79 |
| Table 3.3: Allowable range of concrete crack width based on current design codes with regards to exposure conditions and loading category for reinforced and prestressed concrete structures..... | 81 |
| Table 3.4: Best model performance for CNN classifications for 100 epochs calculation..... | 83 |
| Table 3.5: Comparison between experimental and numerical values for 100 epochs calculation..... | 92 |
| Table 4.1: Adopted 1D CNN Architectures..... | 106 |
| Table 4.2: Adopted 1D CNN Hyperparameters..... | 108 |
| Table 4.3: Structural Parameters for the Three Numerical Models..... | 116 |
| Table 4.4: Comparison between conventional ML and DL Time-Series classifiers using different metrics corresponding to the first damage assessment scenario..... | 120 |
| Table 4.5: BuildingNet_1 Confusion Matrix..... | 122 |
| Table 4.6: BuildingNet_2 Confusion Matrix..... | 124 |
| Table 4.7: BuildingNet_3 confusion matrix..... | 125 |

Table 4.8: Demonstration of Obtained Results for the Three Main DL Models Adopted in this Study 127

List of Figures

| | |
|---|----|
| Figure 2.1: Damage Detection Disciplines. | 10 |
| Figure 2.2: ML Taxonomy..... | 11 |
| Figure 2.3: ML Life Cycle. | 12 |
| Figure 2.4: Input Configuration. | 13 |
| Figure 2.5: List of ML Algorithms Applied to SHM..... | 14 |
| Figure 2.6: SVM Classifier Architecture. | 15 |
| Figure 2.7: Commonly used Configuration for CNN. | 16 |
| Figure 2.8: 3D Model of the Hayakawa Bridge, Japan..... | 32 |
| Figure 2.9: a) 3D model of the Hong-Kong 42 story High-rise Building. b) Scaled Prototype of the Substructures and Location of Sensors along with the Height of the Building..... | 34 |
| Figure 2.10: a) A disposition of the Installed Sensors in a Dam. b) Flow Diagram of DM Data Analysis..... | 35 |
| Figure 2.11: Sensors for WTHM. | 36 |
| Figure 2.12: Sensors Installed in Giuseppe Mazzei Stadium, Italy. | 38 |
| Figure 3.1: Method of Labelling Images According to Their Orientation. | 60 |
| Figure 3.2: Example of Labeled Images According to the Proposed Method..... | 61 |
| Figure 3.3: Overall Architecture of the CNN Classifier. | 62 |
| Figure 3.4: Convolution Operation. | 64 |
| Figure 3.5: Pooling Operation..... | 65 |
| Figure 3.6: Dropout Operation..... | 66 |
| Figure 3.7: Batch Normalization Transform Applied to an Activation x Over a Mini Batch . | 67 |

| | |
|---|-----|
| Figure 3.8: An Example of Image Transformation..... | 73 |
| Figure 3.9: Length Calculation of the Crack. | 74 |
| Figure 3.10: Width Calculation of the Crack..... | 75 |
| Figure 3.11: Configuration of the Angle of Orientation in the Image. | 76 |
| Figure 3.12: Calculation of Angle of Orientation of Crack. | 76 |
| Figure 3.13: Introducing the Area of the Selected Surface..... | 77 |
| Figure 3.14: Causes of Crack Initiation in the Structural Elements of a Frame. | 78 |
| Figure 3.15: Database Inputs Model Incorporation. | 82 |
| Figure 3.16: Accuracy and Loss Histories for the Three Recognition Tasks. | 84 |
| Figure 3.17: Unmanned Confusion Matrix of Test Prediction in the Three Tasks..... | 85 |
| Figure 3.18: Calculation of Crack Features of the Damaged Beam under the effect of Shear Stress. | 86 |
| Figure 3.19: Calculation of Crack Features of the Damaged Beam under the Effect of Flexural Stress..... | 87 |
| Figure 3.20: Calculation of Crack Features of Damaged Pavement under the Effect of uplifting of Tree Roots..... | 88 |
| Figure 3.21: Calculation of Crack Features of a Damaged Reinforced Concrete wall..... | 89 |
| Figure 3.22: An Overview of the Crack Quantification Process. | 90 |
| Figure 3.23: Performance of the Proposed Approach with a Questionable Explanation. | 91 |
| Figure 4.1: BuildingNet Design Methodology. | 103 |
| Figure 4.2: Difference between 1D CNN and 2D CNN. | 105 |
| Figure 4.3: Example of 1D CNN Configuration with 3 CNN and 2 MLP Layers. | 107 |
| Figure 4.4: Three 1D CNN Consecutive Hidden CNN Layers. | 109 |

| | |
|--|-----|
| Figure 4.5: <i>BuildingNet</i> Corresponding to the Three Adopted Case Studies. | 110 |
| Figure 4.6: Example of a Signal Decomposition into N Number of Frames..... | 112 |
| Figure 4.7: Three-Dimensional Model of a Three DOF System. | 117 |
| Figure 4.8: Confusion Matrix Terminology..... | 118 |
| Figure 4.9: Evaluation of BuildingNet_2 Damage Distribution in Terms of Accuracy and f1 score. | 123 |
| Figure 4.10: Comparison of Accuracy Results with 0-20% Random Gaussian Noise Added to the Testing Data. | 125 |

List of Symbols

| | |
|---------------------|--|
| T_{air} | Air Temperature |
| T_{amb} | Ambient temperature |
| H_{up} | Upstream pool level |
| H_{dn} | Downstream pool level |
| T_{Conc} | Concrete Temperature |
| $Precip$ | Precipitation |
| $lag(.)$ | Lagged Variable |
| OL | Output Lag |
| $\partial(.)$ | Derivative of time |
| U_x | Radial displacement |
| Fig | Figure |
| Me | Method |
| E | Experimental |
| A | Algorithm |
| O | Orientation |
| L | Length of the crack |
| L_{ε} | L_{ε} : Length error |
| W | Width of the crack |
| W_{ε} | Width error |
| Ang | The angle of the orientation of the crack |
| Ang_{ε} | Angle error |
| n^L | The height of the image of the convolutional layer L |
| m^L | The width of the image of the convolutional layer L |
| c^L | The number of filters of the convolutional layer L |
| f^L | The size of the filter of the convolutional layer L |

| | |
|-------------------------|---|
| p^L | The padding of the convolutional layer L |
| s^L | The stride of the convolutional layer L |
| f^* | The frequency of the filtered image |
| $Z(i, j)$ | The local gain factor |
| $f_{org}(i, j)$ | The original frequency of the image |
| $f_{blur}(i, j)$ | The frequency of the blurred Image |
| $m1$ | The mean value of the original Image |
| $H(\Omega_i, \Omega_j)$ | The Gaussian Transfer Function |
| σ_0^2 | The probabilities of the two classes |
| σ_1^2 | The variances of the two classes |
| Lb | Number of the bins of the histogram |
| t | Threshold |
| n_c | Width of the resized Image |
| n_r | Height of the Image |
| S_p | The starting point |
| θ | The angle of orientation of the crack |
| x_k^l | The input of the layer L |
| b_k^l | The bias of the k^{th} neuron at layer L |
| E_l | The loss |
| N_t | The number of training data |
| x | The samples of a neural network |
| y | The labels of a neural network |
| a | The output of the neural network |
| Ns | The number of stories |
| N_f | The number of frames |
| M | The mass matrix |

| | |
|---|-----------------------------------|
| C | The Damping matrix |
| K | The stiffness matrix |
| α and β | The Rayleigh damping coefficients |

List of Abbreviations

| | |
|-------------|-------------------------------------|
| ASCE | American Society of Civil Engineers |
| JSCE | Japan Society of Civil Engineers |
| ACI | American Concrete Institute |
| NZ | New Zealand Standards |
| AS | Australians Standards |
| FHWA | Federal Highway Administration |
| SHM | Structural Health Monitoring |
| BHM | Bridge Health Monitoring |
| CC | Construction Control |
| DD | Damage Detection |
| BUHM | Building Health Monitoring |
| DHM | Dam Health Monitoring |
| WTHM | Wind Turbine Health Monitoring |
| WT | Wind Turbine |
| WTB | Wind Turbine Blade |
| AI | Artificial Intelligence |
| DL | Deep Learning |
| TL | Transfer Learning |
| ML | Machine Learning |
| SL | Supervise Learning |
| ANN | Artificial Neural Network |
| NN | Feed Forward Neural Network |
| SVN | Support Vector Network |
| SVM | Support Vector Machine |
| BPNN | Back Propagation Neural Network |

| | |
|---------------|--|
| K-NN | K Nearest Neighbors |
| PCA | Principal Component Analysis |
| CNN | Convolutional Neural Network |
| 2D CNN | Two-Dimensional Convolutional Neural Network |
| 1D CNN | One Dimensional Convolutional Neural Network |
| FCN | Fully Convolutional Neural Network |
| F-RCNN | Faster Region-Based Convolutional Neural Network |
| LSTM | Long Short-Term Memory |
| RNN | Recurrent Neural Network |
| IPT | Image Processing Technique |
| CVB | Convolutional Blocks |
| DCVB | Deconvolutional Blocks |
| FCL | Fully Convolutional Layer |
| VGG | Visual Geometry Group |
| UAV | Unmanned Aerial Vehicle |
| IE | Impact Echo |
| DBM | deep Boltzmann machine |
| HMOO | Hybrid Multi-Objective Optimization |
| GA | Genetic Algorithm |
| EMD | Empirical Mode Decomposition |
| ELM | Extreme Learning Machine |
| MLR | Multiple Linear Regression |
| SWMR | Step Wise Linear Regression |
| GUW | Guided Ultrasound Wave |
| SCADA | Supervisory Control and Data Acquisition systems |
| THM | Tunnel Health Monitoring |
| SOM | Self-Organizing Maps |

| | |
|----------------|---|
| NB | Naïve Bayes |
| AAR | Alkali-Aggregate Reaction |
| DEF | Delayed Ettringite Formation |
| DRI | Damage Rating Index |
| FT | Cyclic Freezing and Thawing (FT) |
| SURF | Speeded Up Robust Feature |
| RGB | Red, Green, Blue |
| ReLU | Rectified Linear Unit |
| NAG | Nesterov Accelerated Gradient |
| DOF | Degree of Freedom |
| NLP | Natural Language Processing |
| MLP | Multi-Layer Perceptron |
| BN | Batch Normalization |
| SMOTE | Synthetic Minority Oversampling Technique |
| MCC | Matthew Correlation Coefficient |
| NLARE | Nonlinear AR Exogenous |
| HNLPCA | Hierarchical nonlinear principal component analysis |
| DT | Decision Trees |
| RF | Random Forests |
| DA | Discriminant analysis |
| RF | Random Forests |
| EC | Ensemble Classifiers |
| NCA | Neighborhood component analysis |
| Softmax | Activation function |

Chapter 1

1 Introduction

1.1 Background

Civil structures and facilities are vulnerable to failure due to structural defects caused by material degradation, earthquakes, wind loadings, ambient vibration, or excessive loading. As reported in (ASCE, 2017), the average rating for USA's roads was a poor D+ on a grade scale of A (excellent condition) to F (unacceptable condition), and C+ for bridges with an approximate \$123 billion needed for retrofitting, as illustrated in (FHWA, 2019).

This latter study reports that 7.5% of bridges were rated structurally defective, with several components near the end of their operational life. Due to their deteriorating conditions, more than 30% of the 617,000 highway bridges in the U.S.A. require urgent consideration.

Ensuring safety and reducing inspection costs have emerged as top priorities for engineering practitioners and researchers in recent decades. This has highlighted the importance of cost-effective structural health monitoring (SHM) to warrant long-term structural integrity and safety levels on several platforms (Mashayekhi and Bell, 2019; An *et al.*, 2019). In addition to traditional inspection and non-destructive evaluation approaches (e.g., use of impact echo, ultrasonic surface waves, ground-penetrating radar, electrical resistivity, infrared thermography, etc.), different forms of new SHM technologies can streamline regular inspections and reduce the direct and indirect costs associated with undesired failure of aging infrastructure assets.

Sensors and sensor data (observable response) lie at the heart of any SHM system and implementation. Recent advances in sensor and communication technologies (contact and contactless, wired, and wireless, etc.) have created opportunities for a tremendous rate and number of observables to be acquired. Furthermore, advancements in other enabling hardware and software were used in diverse ways. Owing to the power and data connectivity specifications, the constraints on sensor measuring technologies and difficulties in implementing sensor networks have traditionally impeded the installation of large sensor arrays on civil infrastructure.

Conventional SHM systems based on proven multi-physics models, however, might not necessarily be appropriate for efficient health monitoring utilizing broad sensor data sets. Fortunately, advancement in data-driven approaches has revolutionized data collection and analysis. Unlike traditional SHM models based on physics, data-driven models offer bottom-up solutions that include diagnosis and prognosis, including damage detection and estimation of the remaining life, respectively (Zhao *et al.*, 2019). Data-driven models have, therefore, proven versatility and become the most attractive strategies in SHM.

Deep Learning (DL) is considered as a sub-branch of machine learning (ML), and its implementations have been adequately demonstrated on several platforms when working with vast volumes of data. DL models can capture and learn information concealed in the data to predict different patterns through stacked-layer blocks that form the DL skeleton (Schmidhuber, 2015). Fortunately, recent developments in parallel computation, coupled with the advancements of DL (Lecun *et al.*, 2015), DL-based models were widely used in many applications in a wide variety of research fields, including computer vision, time series, speech and audio recognition, and SHM. DL-based SHM trained models are better suited to large-scale structures and are more efficient for SHM based on vision and vibration while dealing with compressed or big data.

Different alternate DL models have recently been introduced, such as Deep Convolutional Neural Networks (CNNs) (Rafiei *et al.*, 2017), Deep Boltzmann Machines (Zhang *et al.*, 2017), Deep Belief Network (Zhao *et al.*, 2019), Recurrent Neural Networks (Pathirage *et al.*, 2019), Auto-encoders (Goodfellow *et al.*, 2014), and Generative Adversarial Networks (Bao *et al.*, 2019). In this thesis, the main focus is related to the application of CNNs in SHM, owing to its great success proven in the literature.

1.2 Research Gaps

1. Most modern infrastructure inspection strategies are focused on visual assessment (i.e., presence, location, and width of deterioration and damage) that depend on the skill, experience and knowledge of experts, which may not always be reliable. Furthermore, these strategies are costly, time-consuming, and hindered by requiring access to several sections of complex structures (Kim *et al.*, 2019). A more contemporary image processing

method in the field of image pattern detection called Convolutional Neural Networks (CNN) has gained increasing attention to improve the level of compliance on concrete facilities inspection.

Recent research on applying DL to concrete structures in vision based SHM has focused on using CNNs to classify the different types of structural damage. Nonetheless, few studies have focused their scope on determining the causes of structural harm and the degree of seriousness via strategies centered on DL images. For reference, (Gao and Mosalam, 2018) provided Structural *ImageNet* with four baseline recognition tasks: identification of the component type, spalling condition check, assessment of damage level, and determination of the amount of damage. They identified damage and estimated its degree based on cracks in photos at the structural stage. Another study suggested a fully convolutional neural network (FCN), called *Ci-Net*, for detection of structural crack (Ye *et al.*, 2019). To confirm its structural crack-recognition capability, crack images from an indoor concrete beam study were adopted. On the other hand, a more recent study has suggested an updated, more rapid region-based convolutional neural network (*Faster R-CNN*) for the detection and localization of damaged reinforced concrete columns from images (i.e., concrete cracking, concrete spalling, rebar exposure, and rebar buckling) (Xu *et al.*, 2019).

The various studies listed above-defined structural damage in reinforced concrete members based on traditional usage of DL-based models, which consists of classifying the damage based on a dataset of images of the specified damage class. However, as stated in each analysis, in classifying the type of structural damage occurring on the concrete surface and estimating its extent, there has often been a margin of error and lack of precision. Also, it is a laborious and time-consuming process to label a dataset of images according to their damage shape or pattern to determine the type of harm and its degree. This method depends strongly on engineering judgment and is thus associated with high degrees of subjectivity.

2. Another knowledge gap in the current literature is the dearth of research that addressed the quantification of rational cracks. For example, an automatic volumetric damage quantification (*F-RCNN*) was suggested; it is centered on DL utilizing a depth camera (3D scanner) only to measure the volume of spalling (Beckman *et al.*, 2019).

3. On the other hand, structural engineers reported building collapses in certain situations without any warning (e.g., no noticeable cracking, no spalling, no scaling, no efflorescence, etc.). This can occur due to loss of the rigidity of a beam, column, or other structural elements due to internal degradation, which do not appear on the surface.

4. The existent vibration-based approaches are focused on the assumption that damage (physical adjustments) induces subsequent changes in vibration dynamics (especially modal forms, frequencies, and damping) (Xu *et al.*, 2018), and can be used to determine the position of damage from measured data. Numerous DL algorithms, such as Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), and 1D CNN, have been applied to this nonparametric time series problem. These techniques proved efficient in the case of a network of sensors. For instance, a 1D CNN to track damage in a grandstand simulator was presented, having 30 accelerometers installed on 30 joints (Abdeljaber *et al.*, 2017). In another study, several impact-hammer experiments on a steel frame with six mounted accelerometers in a separate position were performed (Zhang *et al.*, 2019). Such simulations were usually applicable to horizontal systems (e.g., 1.65 m high stand simulator and 1.5 m high steel frame) typically consisting of continuous signal acquisition by a network of accelerometers (multiple channel measurement).

1.3 Research Need and Objectives

According to each of the proposed research gaps cited in Section 1.2, a corresponding research need is suggested and critically discussed in this thesis. Thus, the research objectives of this thesis are to:

1. Develop a novel strategy for determining the type of crack, depending on its orientation to assess the cause and severity of the damage.
2. Carry out accurate quantification of crack characteristics in terms of width, length, and angle of orientation via mathematical and geometric operations to classify structural and durability-related damage of structural members and to determine their extent in short computational time.

3. Propose a more global technique for detecting vibration-based damage for recognizing global structural damage and determining the occurrence, severity, and position of damaged areas by transmitting assessed signals through a network of sensors. DL has developed new horizons for broad-scale structures in vibration-based, data-driven SHM and enabled the collection and processing of large data sets from various sensor types and will be explored for this purpose.

4. Explore utilizing a specific sensor configuration involving single and multiple channel measurements under different damage assessment approaches to track damage to vertical structures (e.g., multi-story buildings) under different damage assessment approaches.

1.4 Original Contributions

This research aims to transform the structural damage prognosis and diagnosis disciplines in concrete structures through the applicability of DL CNNs. Accordingly, two main implementations of CNN were developed. The first is mainly related to the application of 2D CNN for vision based SHM. The second is interested in vibration and signal based SHM using a 1D CNN. Specific original contributions include:

1. Developing a 2D CNN image-based technique with high accuracy and less computing time for automatically classifying crack types in concrete structures based on their orientation since previous studies used manual labeling techniques to identify the type of damage, which is subjective, time-consuming, and laborious. This model quantifies concrete cracks in terms of length, width, and angle of orientation using a combination of DL and improved Otsu image processing technique (IPTs) and identifies the severity of structural damage based on the allowed range of concrete crack widths for different structures, including buildings and bridges, as per guidance from international standards and design codes.
2. Developing a real-time 1D CNN vibration-based technique for assessing damage in mid-rise buildings with high accuracy and short computational time by automating the damage sensitive features extraction pre- and post-processing. The model proposes a single-channel measurement (only one sensor) vibration-based

damage detection platform that can detect and assess the health of a structure, which is more economical and practical. Finally, the performance and robustness of the proposed model was demonstrated by adding up to 20% random Gaussian noise.

1.5 Thesis Structure

This thesis has been structured according to the integrated-article format following the guidelines and regulations of the School of Graduate and Postdoctoral Studies (SGPS) at Western University. The thesis consists of five chapters covering the scope and objectives of the study; to classify, localize, and quantify structural damage in concrete structures using DL CNNs techniques.

Chapter 1 is an overview that captures the context of the study and includes an introduction to the study objectives, research goals, and the original research contributions.

Chapter 2 provides a systematic review of the state-of-the-art with a detailed analysis and discussion of the application of machine and deep learning algorithms in civil SHM. The various ML algorithms used in this domain have been classified into two major subfields: vibration-based SHM and image-based SHM. The efficacy of deploying ML algorithms in SHM has been discussed, and a detailed critical analysis of ML applications in SHM has been provided. Accordingly, practical recommendations have been made, and current knowledge gaps and future research needs have been outlined.

Chapter 3 proposes a nearly automated inspection model based on image processing and DL for detecting defects in concrete structures. The defects are computed in terms of visible cracks on the surface of structural elements. Precise quantification of the crack length, width, and angle of orientation are provided. Furthermore, the type of structural damage and its severity are identified based on the allowed range of concrete crack width for different structures, including buildings and bridges, based on different international standards and codes.

Chapter 4 presents a novel DL-based damage detection approach to automatically extract features from raw acceleration sensor data. A new One-Dimensional Convolutional Neural Network (1D CNN) named *BuildingNet* was designed to learn features and identify damage

locations in real-time under different damage assessment scenarios. Parametric studies were conducted on different layer numbers, numbers of training datasets, and noise levels. An ensemble of systematic studies on the optimization of network architecture and preparation of the training data was performed. Numerical investigations on a midrise building were conducted to demonstrate the accuracy and efficiency of the proposed model framework compared with traditional ML methods. Time-domain monitoring data, both from multiple and single-channel measurements, were used for training and testing three different architectures for *BuildingNet*.

Finally, **Chapter 5** outlines the findings and recommendations of the study and presents suggestions for potential studies.

1.6 References

- Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., & Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, 388, 154-170.
- An, Y., Chatzi, E., Sim, S. H., Laflamme, S., Blachowski, B., & Ou, J. (2019). Recent progress and future trends on damage identification methods for bridge structures. *Structural Control and Health Monitoring*, 26(10), e2416.
- ASCE. ASCE's 2017 Infrastructure Report Card. Retrieved from <https://www.infrastructurereportcard.org/>
- Bao, Y., Tang, Z., Li, H., & Zhang, Y. (2019). Computer vision and deep learning-based data anomaly detection method for structural health monitoring. *Structural Health Monitoring*, 18(2), 401-421.
- FHWA. Bridge Condition by Highway System. (2019). Retrieved from <https://www.fhwa.dot.gov/bridge/nbi/no10/condition19.cfm>
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).
- Kaveh, A., & Dadras, A. (2018). Structural damage identification using an enhanced thermal exchange optimization algorithm. *Engineering Optimization*, 50(3), 430-451.

- Kim, H., Ahn, E., Shin, M., & Sim, S. H. (2019). Crack and non-crack classification from concrete surface images using machine learning. *Structural Health Monitoring*, 18(3), 725-738.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- Mashayekhi, M., & Santini-Bell, E. (2019). Three-dimensional multiscale finite element models for in-service performance assessment of bridges. *Computer-Aided Civil and Infrastructure Engineering*, 34(5), 385-401.
- Pathirage, C. S. N., Li, J., Li, L., Hao, H., Liu, W., & Wang, R. (2019). Development and application of a deep learning-based sparse autoencoder framework for structural damage identification. *Structural Health Monitoring*, 18(1), 103-122.
- Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised Deep Restricted Boltzmann Machine for Estimation of Concrete. *ACI Materials Journal*, 114(2).
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- Xu, J., Fu, Z., Han, Q., Lacidogna, G., & Carpinteri, A. (2018). Micro-cracking monitoring and fracture evaluation for crumb rubber concrete based on acoustic emission techniques. *Structural Health Monitoring*, 17(4), 946-958.
- 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.
- Zhang, T., Biswal, S., & Wang, Y. (2019). SHMnet: condition assessment of bolted connection with beyond human-level performance. *Structural Health Monitoring*, 1475921719881237.
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115, 213-237.

Chapter 2

2 Machine Learning Algorithms in Civil Structural Health Monitoring: A Systematic Review

Applications of ML algorithms in SHM have become of great interest in recent years owing to their superior ability to detect damage and deficiencies in civil engineering structures. With the advent of the Internet of Things, big data, and the enormous and complex backlog of aging civil infrastructure assets, such applications will increase very rapidly. ML can efficiently perform several analyses of clustering, regression, and classification of damage in diverse structures, including bridges, buildings, dams, tunnels, wind turbines, etc. In this Chapter, different ML algorithms used in this domain have been classified into two major subfields: vibration-based SHM and image-based SHM. The efficacy of deploying ML algorithms in SHM has been discussed, and detailed, and critical analysis of ML applications in SHM has been provided. Accordingly, practical recommendations have been made, and current knowledge gaps and future research needs have been outlined.

2.1 Introduction

Civil structures and infrastructures occupy a significant position in the economy and play a vital role in facilitating daily life for the world population. These assets have been incurring premature damage and approaching the end of their service lives (Balageas *et al.*, 2010). Replacing such structures would be costly, labor-intensive, and will exceed available financial and human resources. Hence, engineers have developed various techniques to enhance the safety and structural integrity of those constructions (Karballezadeh *et al.*, 2019) and to mitigate possible financial and life losses associated with their failure. **Figure 2.1** illustrates the different damage detection disciplines in SHM.

This chapter focuses on SHM as a damage detection process. SHM consists of implementing a scheme of monitoring the structure, for instance, using periodically spaced dynamic response measurements, and extracting sensitive features related to damage through these measures and their statistical analyses to assess the actual health of the system (Brownjohn, 2006). Long-term SHM is the result of periodically updated information concerning the ability of the structure to continue serving in the presence of

other influencing factors, such as degradation and aging. Consider, for example, a sudden blast loading (Xu *et al.*, 2018) or a severe seismic event (Limongelli, 2019). SHM could be proposed to provide information on the performance of the structural system during the load event and to assess its structural integrity after that (also termed Rapid Condition Screening) (Agency, 2017). Indeed, SHM can appraise the current state and behavior of a structure via automatically analyzing data acquired by tailored devices and sensors installed in engineered locations across the structure. Hence, anomalies can be duly detected, allowing to instantly assess the reliability of the structure after the catastrophic event, and identifying corrective measures before the damage escalates to more costly or riskier levels.

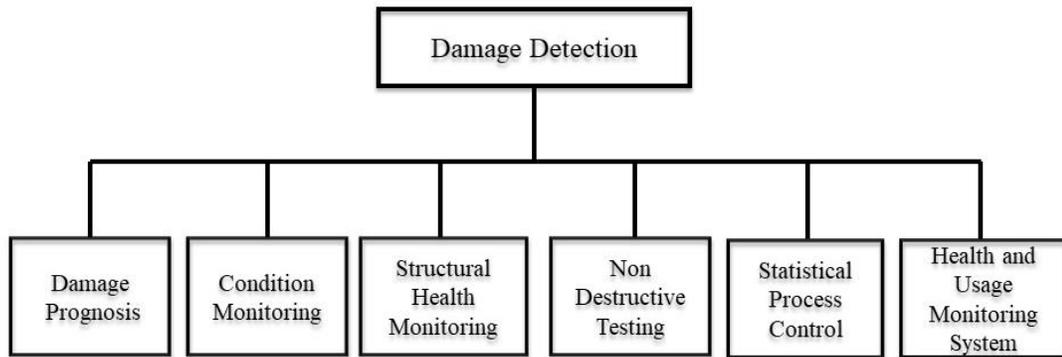


Figure 2.1: Damage Detection Disciplines.

Considering such advantages of SHM, related research has been rapidly escalating and gaining the growing attention of diverse stakeholders. Accordingly, several SHM systems have emerged and been implemented in bridges (Agdas *et al.*, 2015), high-rise buildings (Rafiei and Adeli, 2017), towers (Ochieng *et al.*, 2018), dams (Oliveira and Alegre, 2019), tunnels (Manuello *et al.*, 2019) and so forth. This has led to acquiring big data, which requires powerful, intelligent, and sophisticated computational techniques and has opened the door to deploying Artificial Intelligence (AI) in SHM problems.

Artificial Intelligence emerged between the 1950s and 1970s in the field of computer science and achieved substantial success in various subfields such as robotics (Brooks, 1991), data mining (Wu, 2004), pattern recognition (Pao, 1989), knowledge representation

(Brooks, 1991) and agent systems (Weiss, 1999). Conversely, AI has attracted the attention of civil engineering experts only recently. For instance, it has been used to perform several tasks in SHM applications dealing with knowledge-based systems (Farrar and Worden, 2012), fuzzy logic algorithms (Omar and Nehdi, 2016), and artificial neural networks (Amezquita-Sanchez and Adeli, 2016). The increasing number of AI applications has led scientists and engineers to train more complex models and create more robust AI tools. ML has more recently emerged as a strong contender to deal with this need. It is defined as a subset of AI that uses statistical models to improve the accuracy of machines by understanding the structure of data and then fitting it into models (Farrar and Worden, 2012). A machine could learn via supervised, unsupervised, or reinforcement learning (Figure 2.2).

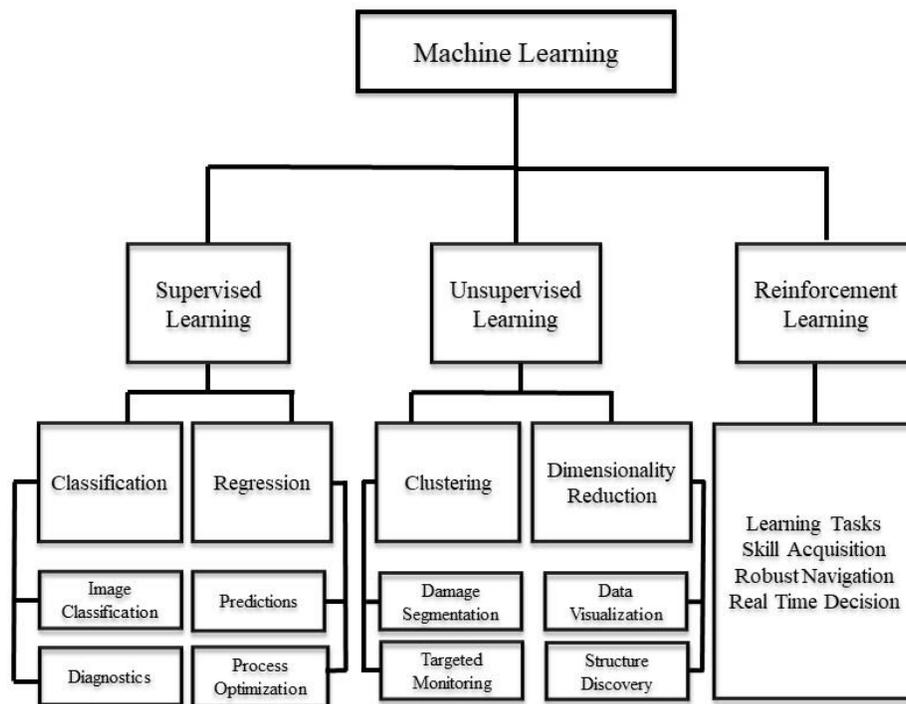


Figure 2.2: ML Taxonomy.

Supervised learning (SL) uses labels or captions so the machine can know the features of the objects added to the labels that are combined with those features. SL provides a learning scheme with labeled data to deal with regression and classification problems. In the SHM domain, SL can be used, for instance, to detect the type and severity of damage (Smarsly

et al., 2016). Conversely, unsupervised learning is the process of learning with unlabeled data, i.e., via datasets with unspecified outputs that fit a general rule and can be grouped following a specific trend. This can be used, for example, to detect the existence of damage through clustering structural response data. As shown in **Figure 2.3**, ML is a straightforward process, starting from the input (Database), passing through the selected algorithm, getting the output, then deciding to either stop or restart the process by providing some feedback. The end of the process is marked by getting an accurate and well-predicted result.

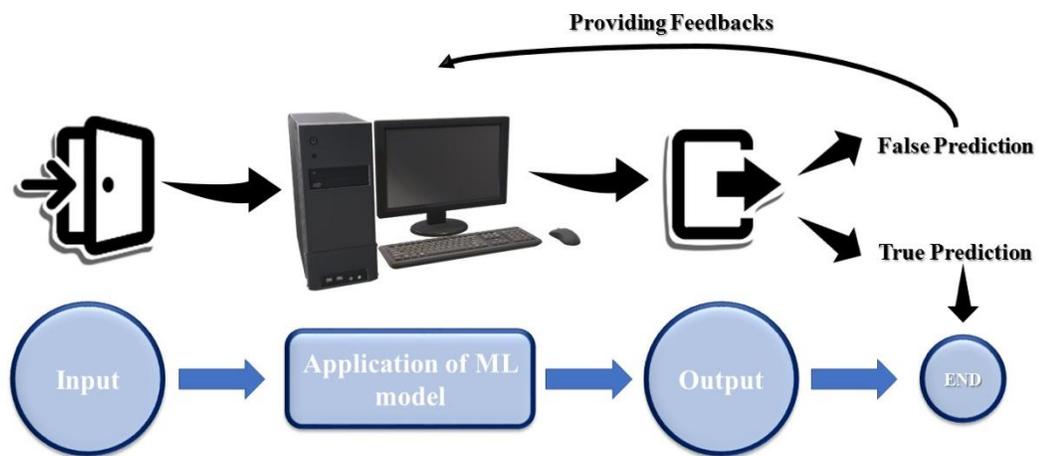


Figure 2.3: ML Life Cycle.

2.2 Hierarchy of ML algorithms

For the sake of clarity, a brief guideline on how to manipulate each of the ML steps of the general process is provided below.

2.2.1 Input Configuration

Starting at the input stage, a better understanding of the data can help in selecting the appropriate algorithm to use. Some algorithms can perform well with smaller sample sets, while others require large samples. Also, some work better with a specific type of data than others. As illustrated in **Figure 2.4**, data need to be well understood and manipulated using mathematical tools such as data statistics and data visualization, before using any ML algorithm. In data statistics, percentiles are used to identify the range, average, and median

of data to describe the central tendency and correlations, besides acquiring knowledge of how the data is linked together (Jordan and Mitchell, 2015). However, in data visualization, density plots and histograms are used to show the distribution of data, along with box plots to identify problems like outliers (Salloum *et al.*, 2019). Then, data need to be ‘cleaned’ which involves dealing with missing values and outliers that can be a concern for some algorithms, decreasing output predictive accuracy. Finally, the data can be augmented or enriched to make the models easier to interpret, reduce data redundancy and dimensionality, capture complex relationships, and rescale some variables.

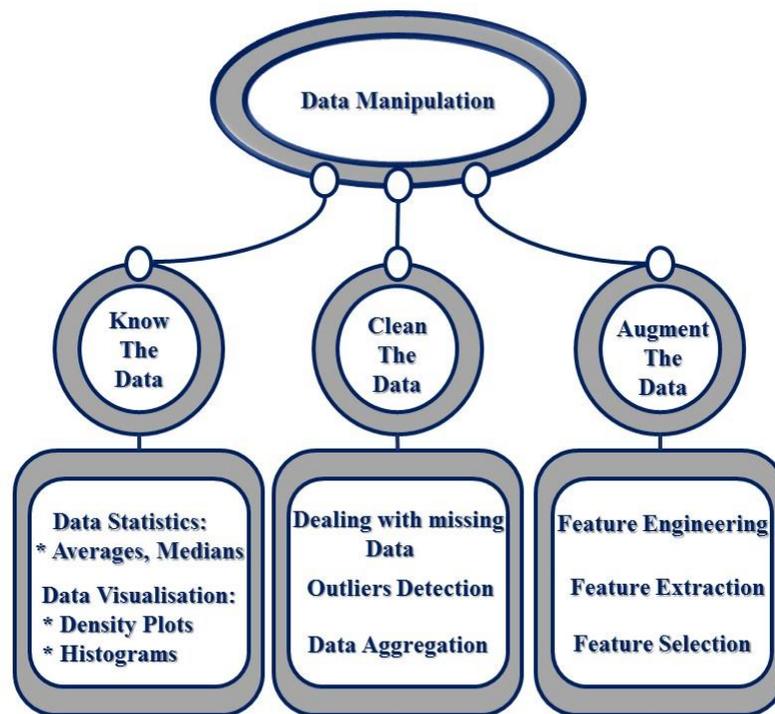


Figure 2.4: Input Configuration.

After manipulating the data, the problem needs to be categorized following an input-output process. For the input process, if the data is labeled, it will consist of a supervised learning problem. However, if it is unlabeled, the learning problem is considered unsupervised. On the other hand, the output process is categorized by the task. If the output is a set of input groups, the problem shall be recognized as a clustering problem. Understanding the constraints of the problem is also a primary task in selecting an appropriate algorithm.

Several kinds of constraints could be presented in an ML algorithm, starting from the awareness of the data storage capacity. Furthermore, the time of prediction can play a significant role in the selection process. For instance, some SHM problems need to be performed promptly. For example, real-time object detection problems need to be super-fast to avoid wasting information during the process of object recognition (de Almeida Cardoso *et al.*, 2019). Besides, the model training process should learn rapidly in cases where it is rapidly exposed to new data and must instantly process it. To select the appropriate algorithm, other factors such as the accuracy and scale of the model, model pre-processing, and complexity in terms of features included to learn and predict more complex polynomial terms, interactions, and more computational overhead, need to be considered. The commonly used ML algorithms in SHM applications are summarized in **Figure 2.5**.

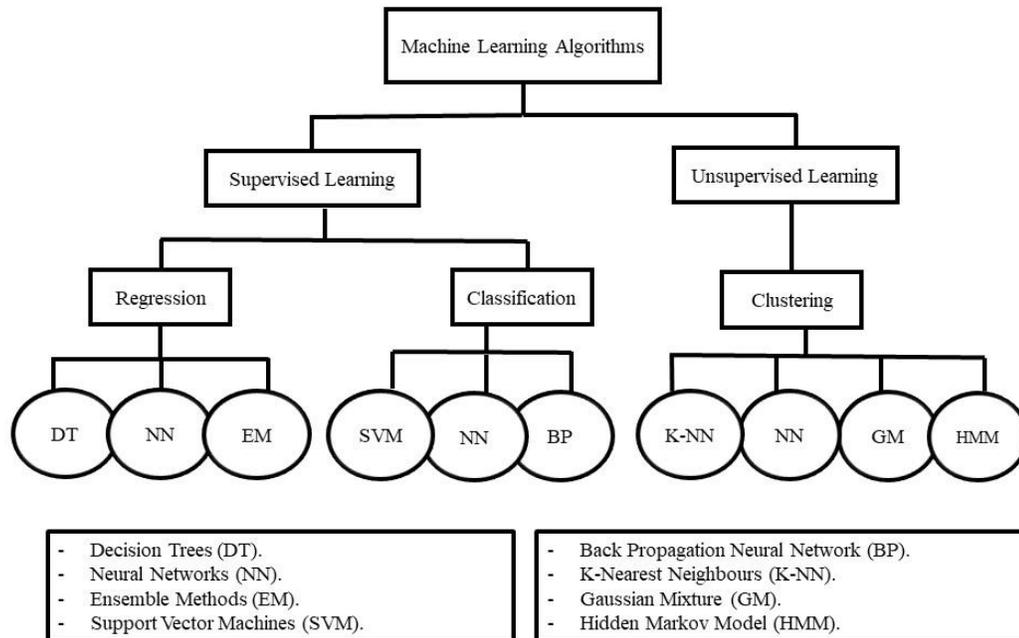


Figure 2.5: List of ML Algorithms Applied to SHM.

2.2.2 Algorithm Manipulation

The most used ML algorithms for SHM purposes are outlined below. Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression

problems, also called Support Vector Networks (SVN). An SVM algorithm sorts data into one of two categories, then outputs a map of the sorted data, maximizing the margins between the two. It performs both linear and non-linear classifications, thanks to the use of kernel functions (Burges, 1998). Its architecture is detailed in **Figure 2.6**.

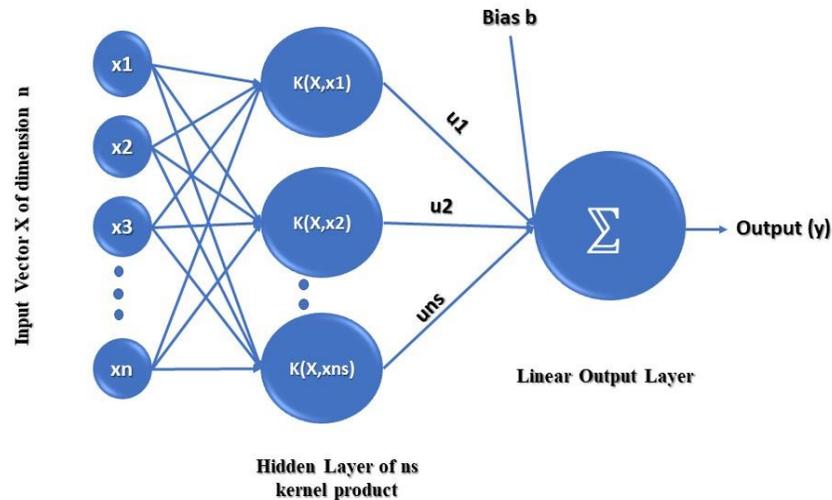


Figure 2.6: SVM Classifier Architecture.

Back Propagation Neural Networks (BPNNs) are supervised learning algorithms for training multi-layer perceptrons. Its primary use consists of finding the minimal value of the error function in the weight space using a gradient descent technique. The weight that minimizes the loss function is the solution for the learning problem (Hecht-Nielsen, 1992). K-Nearest Neighbors (K-NNs) are a set of classifiers used for pattern classification and ML (Dudani, 1976). For a set of inputs x of n points and a distance function, KNNs search for the closest points in x to a query point or set of points y to be found. Principal Component Analysis (PCA) is a method within the data analysis family that consists of transforming correlated variables to uncorrelated ones, called principal variables. This technique helps the user reducing the size of variables and making the information less redundant (Jolliffe, 2011). CNN is an architecture used in DL, which is a subset of ML, to perform both descriptive and generative tasks dedicated mainly to image processing tasks

using machine vision libraries that contain image and video recognition scripts. The main difference between the ML and DL processes is the hidden layer located between the input and output for DL algorithms, as illustrated in **Figure 2.7**. This layer can contain multiple convolutional or deconvolutional layers, pooling, activation, fully connected, and normalization layers, depending on the use.

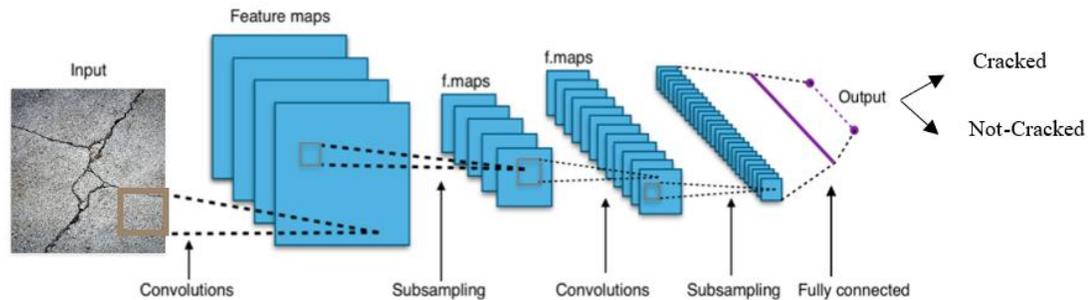


Figure 2.7: Commonly used Configuration for CNN.

2.2.3 Output Manipulation

The output of the SHM can vary from one problem to another, such as settlement, damage detection, damage classification, object detection, temperature prediction, and health index. An accurate and precise output should mark the end of the process as otherwise, feedback is provided to the machine, so it can learn from the experience and attempt to provide better results.

2.3 SHM Subfields

2.3.1 Bridge Health Monitoring (BHM)

BHM is the application of SHM and inspection techniques to bridge structures. Causes of degradation of bridge structures include materials aging (Hasni *et al.*, 2017), corrosion of metals (Zajec *et al.*, 2018) and structural supports (Zhao *et al.*, 2014), mechanical overloading and other damage mechanisms (Chen and Ni, 2018). Bridge Health Monitoring (BHM) consists of collecting quantitative data from various sensors located within or on the surface of the structure (Hao *et al.*, 2018). This Real-Time feedback

creates a dataset monitoring system used to assess the condition of the bridge. Processing real-time complex big data has been a challenge in BHM. According to (Peng *et al.*, 2017), BHM can be separated into three key aspects. First, the construction control (CC) stage, where engineers are responsible for monitoring construction progress. Second, the routine monitoring (RM) stage directly after constructing the bridge. In this period, a large amount of data acquired from the installed sensors is produced and stored. To process this data, ML algorithms are being developed to provide real-time feedback for understanding the health condition of the bridge. Finally, the damage detection (DD) stage where engineers should assess the safety of the structure and detect any damage that develops.

2.3.2 Building Health Monitoring (BUHM)

Buildings are often exposed to damage from earthquakes, wind, overloading, vibration, impact, landslides, floods, aging and environmental action, and other damage mechanisms. Without adequate monitoring, maintenance, and repair, this can lead to inadequate service and possible economic and life loss. Thus, understanding how buildings perform in real conditions can help engineers designing and building more resilient, safer, reliable, and more durable structures. There has been recently rapid growth in the construction of high-rise buildings that require smarter and more robust monitoring (Ali and Al-Kodmany, 2012). Monitoring the deformation of such buildings has long been a concern. More recently, experts have introduced ML algorithms to monitor the condition of high-rise buildings considering their proven effectiveness in other fields.

2.3.3 Dam Health Monitoring (DHM)

Dams play a crucial role in providing drinking and irrigation water, flood defense, power generation, water storage, and so forth. Their deterioration can lead to massive financial losses and possibly a disastrous number of casualties (Brown and Graham, 1988). Thus, the safe operation of dams is needed, and any anomalous behavior should be detected in its early stages to avoid any failure or mis-operation. Dam Health Monitoring (DHM) is a discipline that is often based on a traditional visual inspection and other monitoring of the dam and foundation (Dams, 2012). This requires a robust analysis of dam monitoring data obtained from the installed sensors in the short- and long-term. For short term monitoring,

the engineer is responsible for comparing the measured data with reference values that correspond to the response of the dam to loads in a healthy or safe condition. The detection of anomalies is marked by the localization of predicted intervals located either above or below the reference values. However, for long-term monitoring, analysis of the behavior models and the observed data is needed to assess the performance of the dam in terms of loads and observed output (Kang *et al.*, 2019). DHM can also consist of static and dynamic monitoring aspects. Statically, many features could be monitored, including reservoir storage levels, cracks, displacements, strains, and stresses. Dynamically, other parameters could be identified, like the stiffness, damping ratio, and mode shapes caused by wind, water waves, and ground motions (Fisher *et al.*, 2017). The structural behavior of dams has complicated relationships with environmental factors, hydraulics (e.g., water level) and geo-mechanisms (e.g., pore pressure, rock deformability) (Gunn, 2015). To illustrate the behavior of the concrete dams based on real-time monitoring, several mathematical models have been proposed, including statistic, deterministic and hybrid models. Such models serve to assess the behavior of dams by analyzing real-time data, considering hydrostatic pressure, environmental temperature, and time effects to be the main variables (Su *et al.*, 2015). Due to uncertainties in using this kind of approach, several AI techniques have been implemented, making a fusion between conventional models and heuristic algorithms, and leading to hybrid models. In recent years, ML has become a new accurate tool in DHM.

2.3.4 Wind Turbine Health Monitoring (WTHM)

To limit the need for traditional sources of energy such as fossil fuels, eco-friendly sources of energy that can mitigate climate change are being sought after (Hadjipaschalis *et al.*, 2009). Wind Turbines (WT) have gained acceptance owing to the maturity of their technology. More significant size WT emerged to harvest more wind energy, seeking efficiency and productivity. However, this reason has complicated maintenance and repair works for facility managers. Several attempts to monitor the structural integrity of WT have been reported. For instance, different problems faced by wind turbine blades (WTB) during their lifecycle (Ciang *et al.*, 2008), and methods used to detect damage in WT, including acoustic emission event detection (Sutherland *et al.*, 1994), thermal imaging (Avdelidis *et al.*, 2006), ultrasonic methods (Sørensen *et al.*, 2002), modal based

approaches (Siringoringo and Fujino, 2006), fiber optics (Takeda, 2002), laser doppler vibrometer (Martarelli *et al.*, 2001), electrical resistance-based damage detection (Matsuzaki and Todoroki, 2006), strain memory alloy (Verijenko and Verijenko, 2005), X-radioscopy (Sørensen *et al.*, 2002), eddy current (Gros, 1995), and other methods have been reported. Accordingly, big data have been cumulated. Data science is needed for classification and prediction of WT damage, hence the need for ML.

2.4 DL and ML Applications in SHM

This section surveys different ML and DL approaches and algorithms used in SHM problems. Various algorithms were used in SHM applications for the last ten years, including Back Propagation (BP) algorithm, SVM, Neural Networks (NNs), K-Nearest Neighbors, CNNs. Uses of those algorithms in several applications, including SHM of bridges, high-rise buildings, dams, and wind turbines, are outlined below.

2.4.1 Artificial Neural Networks (ANNs)

2.4.1.1 Feed Forward Neural Networks (N.N.s)

Gonzalez *et al.* (González and Zapico, 2008) presented a damage identification method for steel moment frame structures. The method uses NNs and first flexural modes (frequencies and mode shapes obtained by a finite element model for a five-story office building) as input. Their method was based on two main approaches. The first is to calibrate the healthy structure, while the second was intended to identify the damaged structure after a seismic event. They predicted the mass and stiffness of the structure to provide a damage index at each story and indicated a robust model prediction of damage. More recently, Chang *et al.* (Chang *et al.*, 2018) developed this approach and applied it not only to detect damage but also to localize it and predict its severity for appraising the remaining performance of the damaged members. Two critical structures were studied: (i) a seven-story building with single and multiple damaged columns, and (ii) a scaled twin tower with weak braces installed on some floors.

To detect damage (DD) in bridges, three different algorithms were applied. The NN technique was used in the Jamboree road over-crossing, Irvine, California, to assess

parameters including aging, long-term structural parameters, stiffness, and mass (Soyoz and Feng, 2009). Many applications have used this algorithm owing to its simplicity and accuracy compared to traditional methods. For instance, it was used to determine radial dam displacements with different sets of inputs (Salazar *et al.*, 2015; Riquelme *et al.*, 2011; Kao and Loh, 2013; Demirkaya and Balcilar, 2012; Mata, 2011). Other uses were reported in (Simon *et al.*, 2013; Ranković *et al.*, 2014; Nourani and Babakhani, 2012) to detect the pore pressure in dams, to predict the tangential displacement (Popovici *et al.*, 2013) and to monitor the leakage flow (Santillán *et al.*, 2014). A summary of the used algorithms is provided in **Table 1.1**.

2.4.1.2 Back Propagation Neural Networks (BPNNs)

BP algorithm was applied during the early stages of construction of the Yangtze River bridge in China to track girder elevation changes during the construction phase using input parameters like cable tension deflection parameters and deflection of the deck. Another study (Peng, Zhang, Peng and Liang, 2017) employed a BP algorithm to track variation of the deflection of the Hubei Danjiangkou bridge deck throughout the Construction Control (CC) phase, using inputs including temperature, the value of deflection of the deck after stretching and height of the stretched section. Other uses of the BP algorithm were in the Routine Monitoring (RM) stage. For instance, pile settlement was predicted as a function of the pile displacement sequence (Peng, Zhang, Peng, and Liang, 2017) and to track the normality of points according to their deflection (Yang *et al.*, 2008). The Kentucky Louisville truss bridge in the USA was exposed to an extensive campaign to measure parameters like frequency, mode shapes and the number of degrees of freedom to serve as inputs for measuring the damage potential of truss joints (Mehrjoo *et al.*, 2008; Frangopol and Soliman, 2016). The Yangtze River Bridge was also monitored to track girder elevation changes based on cable tension and deflection parameters using BPNN, as illustrated in (Yuansong *et al.*, 2007).

Table 2.1: Summary of the Different NN Applications in SHM.

| Ref. | Structure | Input | Algorithm | Output |
|--|--|--|------------------|---|
| (González and Zapico, 2008) | Five-story steel office building | Frequencies and mode shapes | NN | Mass and stiffness degradation for damage index detection |
| (Chang, Lin and Chang, 2018) | 1- Seven story building with single and multiple damaged column(s). 2- Experimental model of a scaled twin-tower building with weak braces in some floors | Modal properties of the structure under ambient vibrations | NN | Damage patterns in terms of stiffness reduction after critical events |
| (Soyoz and Feng, 2009) | Jamboree Road over-crossing, Irvine, California, USA | Modal parameters: Frequencies, Mode Shapes | NN | Aging and Long-term structural, parameters, |
| (Salazar, Toledo, Oñate and Morán, 2015) | Arch Dam, La Baells, Spain | H _{up} , T _{air} , $\partial(H_{up})$, Season, Time, Precip | NN | Rad_Dis, Tan_Dis, Leakage flow |
| (Riquelme, Fraile, Santillán, Morán and Toledo, 2011) | Arch Dam, La Baells, Spain | H _{up} , T _{amb} , OL | NN | Rad_Dis |
| (Kao and Loh, 2013) | Arch Dam, Fei-Tsui, Taiwan | H _{up} , T _{Conc} | NN | Rad_Dis |
| (Demirkaya and Balcilar, 2012) | Arch Dam, Schelegeis, Austria | H _{up} , T _{air} , T _{conc} , lag (T _{air}), lag (T _{conc}) | NN | Rad_Dis |
| (Simon, Royer, Mauris and Fabre, 2013) | Arch Dam, Pareloup, France | H _{up} , Season, T _{amb} , T _{air} | NN | stiffness, mass, Rad_Dis |
| (Ranković, Novaković, Grujović, Divac and Milivojević, 2014) | (Earth fill + Gravity Arch) Dam, Iron Gate 2, Serbia/Romania | H _{dn} , lag (H _{dn}) | NN | Pore pressure |
| (Nourani and Babakhani, 2012) | Earth fill Dam, Sahand, Iran | H _{up} , H _{dn} , Precip, lag (Precip) | NN | Pore pressure |
| (Popovici, Ilinca and Ayvaz, 2013) | Buttress Dam, Gura Raului, Romania | Time, H _{up} , T _{air} | NN | Rad_Dis and Tan_Dis |
| (Santillán, Fraile-Ardanuy and Toledo, 2014) | Arch Dam, La Baells, Spain | H _{up} , T _{air} , $\partial(H_{up})$, $\partial(T_{air})$ | NN | Leakage flow |
| (Mata, 2011) | Arch Dam, Alto-Rabagao, Portugal | H _{up} , Season | NN | Rad_Dis |

Four distinct uses of ML to detect damage and identify its degree for the main structural elements of a building using the BP algorithm were reported in (Fan *et al.*, 2015). The first consisted of identifying the damage of a reinforced concrete frame structure using the changing ratio of modal strain energy, which is taken as the damage location factor. The second explored damage location and a degree in a simply supported beam, coupled with finite element simulation to calculate the first two natural frequencies of the structure using curvature mode of some critical points highlighted in the frame. The third application identified the damage degree in a scaled four-story steel frame structure where the inputs of the algorithm consisted of ratios of natural frequency, while the applied load was simulated to wind load. Finally, a damage identification method was applied to the Kewitte single-layer spherical reticulated shell. The above methods achieved adequate accuracy in detecting damage for different kinds of structures (**Table 2.2**).

2.4.1.3 Convolutional Neural Networks (CNNs)

More recently, DL (LeCun *et al.*, 2015) has emerged as a sophisticated subset of AI. It has been proposed to perform more advanced tasks using innovative algorithms. Its main application for SHM is detecting defects such as cracks, efflorescence, steel exposure, rust staining, scaling, spalling of concrete structures based on surface images, fatigue in steel structures, bolts loosening, potholes and holes in asphalt pavement, etc. ML allows detecting cracks in civil engineering structures in a fast and reliable way, determining the type of the crack, its distribution along the section, and its width and length. Thus, engineers can assess the load-carrying capacity and degradation level of structures (Shan *et al.*, 2016).

This procedure has often been conducted by experts (Dhital and Lee, 2012) based on rather subjective opinions in assessing the health of structures (Fujita and Hamamoto, 2011) and predicting remaining service, which is compounded by difficulty accessing hard to reach areas. Thus, there is a need for automated and intelligent crack detection methods that do not rely on subjective operator expertise and opinion. Recently, a new technology of automatic crack detection using DL has emerged.

Table 2.2: Summary of the Different BPNN Applications in SHM.

| Ref. | Structure | Input | Algorithm | Output |
|---|--|---|------------------|--|
| Peng <i>et al.</i> , 2017 | Hubei Danjiangkou Bridge | Temperature, Deflection after stretching, Height of stretched section | BPNN | Deflection variation |
| | Beijing-Shanghai High Valence Kunshan Iron Bridge | Pile Settlement Displacement Sequence | BPNN | Prediction of pile settlement |
| Yang <i>et al.</i> , 2008 | Masangxi Bridge | Deflection of points | BPNN | Normality of points |
| Mehrjoo <i>et al.</i> , 2008 Frangopol and Soliman, 2016 | Kentucky Louisville Bridge | Natural frequency | BPNN | Damage Potentials |
| | | Number of modes | | |
| | | Number of the measured Degree of Freedom | | |
| Fan <i>et al.</i> , 2015 | Steel Frame | Changing the ratio of modal strain energy MSECR | BPNN | Damage detection of frame structures |
| | Finite element simulation of the first mode shapes | Vibration signals, Natural frequencies, Mode Shapes | BPNN | Damage position and degree for simply supported beam |
| | Four-story steel frame structure experimental 3D model | Natural frequency change ratios, simulated Wind load | BPNN | Damage degree identification |
| | Spherical reticulated Shell structure | Modal Density, Number of degrees of freedom | BPNN | Damage degree identification |
| Yuansong <i>et al.</i> , 2007 | Yangtze River Bridge | Cable tension deflection parameters, Deck deflection | BPNN | Girder elevation Changes |

New optimization of pre-trained networks such as GoogleNet (Simonyan and Zisserman, 2014), AlexNet (Alom *et al.*, 2018), ResNet (Wu *et al.*, 2019), VGG-16 (2019), YOLO object detection (Redmon *et al.*, 2016) are frequently reported. However, from Input or dataset to output, parameters need to be carefully considered. A summary of the most recent applications of CNNs to detect damage in concrete and non-concrete structures is provided in **Table 2.3** and described below.

It is widely accepted that the more extensive and more comprehensive is the data set, the more successful it can be AI models using such data. Thus, some techniques such as data augmentation (Fawzi *et al.*, 2016) have been proposed to solve problems of lack of data, and to reduce overfitting caused by limited and imbalanced training datasets. Another promising technique that helped increasing prediction accuracy is the dropout technique, which consists of randomly and temporarily ignoring in calculations some units of the neural network. Also, to obtain higher accuracy in image data processing, several parameters should be considered, such as uncontrolled image shooting distance (Snell *et al.*, 2017), lighting conditions (Wang *et al.*, 2017), shot angle, and blurriness conditions.

Most relevant studies have focused on classifying structures as damaged or not damaged through the presence of cracks. One of the earliest applications of CNNs used different layout and architectures, varying the number of convolutional blocks, pooling layers, fully connected layers, adding some features to the available pre-trained networks Transfer Learning (TL) in order to detect cracks in concrete structures and asphalt pavements (Yang *et al.*, 2018).

Different configurations have been proposed to optimize crack detection in defective structures. Recently, a new robust concept based on transfer learning to early detect fatigue cracks in gusset plate joints of steel bridges was proposed in (Dung *et al.*, 2019) as an alternative for training a neural network. They used the output features of the VGG16 network architecture previously trained using a dataset called ImageNet, then they fine-tuned the top layer of VGG16, which helped to achieve the best precision.

Table 2.3: Summary of CNNs applications for SHM.

| Ref. | Number of Images | Pretraining Dataset | Image Preprocessing | DL Algorithm | Topology | Parameter Estimation | Location |
|-----------------------------|-------------------------|--|---------------------------------|--|--------------------------|---|--|
| Wang <i>et al.</i> , 2017 | 332 | ImageNet | Dropout Technique | CNN | 4CVB + 1 ReLu + 1Softmax | Existance of Cracks | Complex Engineering Building |
| Maeda <i>et al.</i> , 2018 | 9053 | MS-COCO | Similar to Pascal-VOC | SSD Using Inception V2, and Mobile Net | | 8 Damage Types including Cracks, Rutting, pothole, etc., | Road Cracks with the cooperation of some municipalities |
| Gao and Mosalam, 2018 | 10000 | ImageNet | Feature Extraction, Fine-tuning | TL based on VGGNet | 5CVB, 1FCL | Component Type, Spalling condition, damage level, damage type | Collected from various platforms: NISEE, NEEShub, EERI Learning, Google Image, Baidu Image |
| Yang <i>et al.</i> , 2018 | 400 | Manually Labelled | Fine-tuning | CNN | 4CVB | Existence of cracks | Asphalt pavement images collected during field survey of several road sections in Da Nang city |
| Kim and Cho, 2018 | 12379 | Manually Labelled, use of Scrape box [http://www.scrapebox.com/] | Data Augmentation | CNN Based on AlexNet | Same as AlexNet | Existence of cracks, Plants, Edge detection | Collected from the Internet covering five kinds (intact |
| | | | | | | | surfaces, cracks, multiple joints, and edges, single joint or edge) |
| Dorafshan and Maguire, 2018 | 3420 | ImageNet | Noise Filtering | CNN based on AlexNet | Same as AlexNet | Existence of cracks and their density | Structural Health laboratory (SMASH Lab) at Utah State University |

Table 2.3 (Continued)

| Ref. | Number of Images | Pretraining Dataset | Image Preprocessing | DL Algorithm | Topology | Parameter Estimation | Location |
|----------------------------------|------------------|-----------------------------------|--|--|--------------------------|--|---|
| Dung <i>et al.</i> , 2019 | 327 | ImageNet | Data Augmentation | SCNN, BN, FT | 2CVB + 1FCL | Existence of Cracks | Gusset Plate welded joints of steel bridges in Tokyo City |
| Huthwohl <i>et al.</i> , 2019 | 38408 | ImageNet | Cross Learning Strategy - Fine-tuning | Inception V3 | | Cracks, Efflorescence, Exposed Reinforcement, Rust Staining, Scaling, Spalling | Walls, Beams, and columns of Concrete Bridges |
| Lee <i>et al.</i> , 2019 | 242 | MS-COCO | 2D Gaussian kernel - Brownian motion process - Data Augmentation | CSN (Image Segmentation Network), CNN Patch Based | 5CVB + 1TCVB + 4DCVB | Existence of Cracks | Around the university of Cambridge Campus |
| Liu <i>et al.</i> , 2019 | 537 | Manually Labelled | Data Augmentation | DeepCrack: FCN + DSN | 5CVB + 4DCVB | Existence of Cracks | Concrete and Asphalt Cracks |
| Murao <i>et al.</i> , 2019 | 552 | Manually Labelled | Data Augmentation | YOLO-v2 | | Existence of Cracks | Crack Image of the concrete wall located at the campus of Kansai University |
| Kim <i>et al.</i> , 2019 | 487 | Manually Labelled | Image Binarization and Noise Removal | SURF-based classification and CNN-based classification | | Existence and Location of Cracks | Random Images of defected structural elements presenting Cracks, Spalling, Holes. |
| Li <i>et al.</i> , 2019 | 2750 | Dense-Net 121 | Data Augmentation, Dropout Technique | FCN | 11CVB + 6DCVB + 1Softmax | Cracks, Efflorescence, Spalling | |
| Zhang <i>et al.</i> , 2019 | 300 | | Format Factoring, unified jpg Format | Faster RCNN, VGG16 | Same as VGG16 | Bolts Loosening | Experiment Structure containing intact and loosened bolts |
| Protopapada <i>et al.</i> , 2019 | 200 | Data Augmentation, Image resizing | Noise Filtering | CNN | 3CVB + 1FCL | Existence of Cracks | Tunnels of Egnatia Motorway in Metsovo |

Table 2.3 (Continued)

| Ref. | Number of Images | Pretraining Dataset | Image Preprocessing | DL Algorithm | Topology | Parameter Estimation | Location |
|------------------------------|------------------|--|--|--------------------------|------------------------------|--|---|
| Zhang <i>et al.</i> , 2019 | 500 | ImageNet | Unified PASCAL VOC format | RCNN Based on ResNet 101 | Same as ResNet 102 | Efflorescence and spalling | The Palace Museum was the Imperial Palace of the Ming and Qing dynasties in China |
| Liang, 2019 | 350 | Manually Labelled | Data Augmentation | CNN | 5CVB + 1FCL + 1Softmax | System-level failure classification, component-level bridge column detection, and local damage-level damage localization | Related research reports on RC bridges and search engines (e.g., Google Image). Post-earthquake images of damaged RC bridges around the world, images of RC columns in different experimental studies |
| Beckman <i>et al.</i> , 2019 | 1091 | Manually Labelled LabelImg [https://github.com/tzutalin/labelImg] | Noise Filtering | Faster RCNN | 5CVB + 1FCL + 1Softmax | Damage Extraction, Volume Quantification | Collected from Internet |
| Ni <i>et al.</i> , 2019 | 163 | ImageNet | Image Inpainting, Image Localization, Feature Extraction | Dual-scale CNNs | Same as GoogleNet and ResNet | Crack localization, Crack width | Laboratory and outdoor |
| Kim <i>et al.</i> , 2018 | 384 | Cifar-10 dataset | Cropping, Quantifying | RCNN | 3CVB+2FCL +1Softmax | Crack localization, Crack width Crack Length | Real Bridge UAV image acquisition |

This affirmed that fine-tuning a well-trained, fully connected layer with the top convolutional layer of the VGG16, in combination with data augmentation, is among the best performing combinations for detecting cracks in structures. Numerous applications have been proposed in the literature looking for the most robust algorithm for cracks detection (Wang *et al.*, 2017; Liu *et al.*, 2019; Lee *et al.*, 2019; Murao *et al.*, 2019; Kim *et al.*, 2019; Li *et al.*, 2019; Zhang *et al.*, 2019; Protopapadakis *et al.*, 2019; Wang *et al.*, 2019 and Dorafshan *et al.*, 2018) through varying the architecture of the used CNN, changing the number of convolutional blocks, which varied between two (Dung, Sekiya, Hirano, Okatani and Miki, 2019) and eleven (Li, Zhao, and Zhou, 2019) convolutional blocks, introducing more pooling at the end of each convolutional block, more activation layers, and normalization, etc.

Other research efforts did not limit their scope to the binary classifications of structure (cracked or not). More innovative and useful ideas for monitoring tasks, for instance, to detect efflorescence and spalling (Li, Zhao, and Zhou, 2019; Hüthwohl *et al.*, 2019); bolts loosening (Zhang *et al.*, 2019), rutting of asphalt pavements and potholes (Maeda *et al.*, 2018), typology of cracks, their length and width (Yang *et al.*, 2018) have been explored. For instance, (Hüthwohl *et al.* 2019) proposed a three-staged concrete defect classifier that can classify unhealthy defected bridge areas and determine their specific defect type compared to inspection guidelines. The process consisted of finetuning three separate pre-trained networks on a multi-source dataset for concrete walls, beams, columns, etc.

Another successful application of CNN was discussed in (Gao and Mosalam, 2018), which proposed a baseline recognition task that determines the component type, check the spalling condition, evaluates damage in percentage (no damage, minor damage, medium to severe damage, collapse) and predicts the mechanical source of damage; e.g., if the crack is horizontal, the mechanical force that initiated it is an axial (tensile or compressive) force; however, if the crack is slightly vertical, a bending moment could be the leading cause; and finally if the crack is inclined, the shear force would be the leading cause. Accordingly, a dataset composed of 10000 images was collected from a platform called ImageNet and then labeled manually for specified recognition tasks. To avoid overfitting, TL based on

VGGNet was applied using two different strategies called finetuning and feature extraction. Two sets of experiments were done to find the relative optimal model parameters and hyperparameters, including learning rate, mini-batch size, number of epochs, initial weights, etc. Both strategies proved useful in recognition applications.

Similarly, a study conducted by (Liang, 2019) proposed a three-level image-based approach for post-disaster monitoring of reinforced concrete bridges using image classification, object detection, and semantic segmentation, respectively to assess the failure of the overall system, detect the structural element (Deck, Column, Beam, Wall) where the damage persists and then zoom to the exact location on that element to localize the damage. This study achieved over 90% accuracy for the three DL models, which confirms the necessity of research to propose new solutions for these kinds of problems.

DL and CNN scholars did not limit their scope in the field of image recognition and attempted diverse applications to detect crack damage in real-time for instance using unmanned aerial vehicles or drones, as illustrated in (Maeda *et al.*, 2018; Kim *et al.*, 2017; Kim *et al.*, 2019; Chen and Jahanshahi, 2017). Collecting images and labeling them manually can be a repetitive and time-consuming task. For this reason, different methods have been used in the literature to save time and provide an alternative solution, such as the use of Scrapebox proposed in (Kim and Cho, 2018), which scrapes images from a search engine site (e.g., Google Images, Baidu Images, etc.) for a keyword (e.g., concrete crack), and LabelImg used as a graphical image annotation tool (in Beckman *et al.*, 2019).

Only a few applications of CNNs have quantified detected cracks on images by calculating its width and length. For instance, (R-CNN)-based transfer learning was applied to 384 collected images (in Kim *et al.*, 2018). Those images were cropped to regions where the crack had been located. To quantify cracks, the exact pixel size in the image and the focal distance were attributed using GPS data of the Unmanned Aerial Vehicle (UAV) system. The crack quantification algorithm was verified in a small-scale laboratory test that provided a relative error of 1~2%. Another application (in Ni *et al.*, 2019) proposed a DL-enabled quantitative crack width measurement method. The study presented a novel crack

width estimation method based on the use of Zernike moment operator, which achieved high accuracy for thin cracks.

2.4.2 Support Vector Machine (SVM)

SVM has been widely used in BHM applications, for instance, to determine damage in the Hangzhou bridge using strain vibration, distortion, and cable tension (Chongchong *et al.*, 2011). For the Flushing 149th bridge in New-York, Impact Echo (IE) data was collected to classify damage of the deck using SVM (Li *et al.*, 2017). Moreover, an attempt was made to use SVM for crack detection in the Sydney Harbor Bridge, Australia, using inputs including force, acceleration, and time histories recorded during regular bridge operation (Alamdari *et al.*, 2016). The SVM algorithm was used in the RM stage, for example, in the Humboldt bay middle channel bridge to evaluate the correct position of the pier using some pier features. To predict scour depth near the bridge piers of the Taiwan High-Speed Rail System Bridge, features like pile length, young's modulus of soil, and natural frequency of the bridge were used with an SVM algorithm (Kerh and Ting, 2005).

To detect and localize damage, two potential applications for SVM have been reported. The first (Li *et al.*, 2018) used a radial basis function for regressing and optimizing the input (mode curvature change). Excellent accuracy and generalization ability, along with noise resistance from the surrounding environment, were achieved. In the second, (Oiwa *et al.*, 2017) applied SVM algorithm to vibration signals from sensors installed on a wooden brace inside a wooden house (Timber Health Monitoring) to track the degradation of wood, assess and localize damage, then compare results to that of k-Nearest Neighbors algorithm. SVM was found more accurate and gave more precise results than the K-NN algorithm for this kind of application. Two main other applications consisted of calculating tangential displacements of the Iron Gate two dams between Serbia and Romania using the downstream height, upstream height, their lags, and the lag of the output itself for next iterations (Ranković *et al.*, 2014). This was intended to predict radial displacements (Rad-Disp) and uplift pressure (Cheng and Zheng, 2013). Also, an evaluation of the correct position of piers installed in the Humboldt bay middle channel California bridge as illustrated (in Bulut *et al.*, 2005). The various SVM applications are summarized in **Table 2.4**.

Table 2.4: Summary of the Different SVM Applications in SHM.

| Ref. | Structure | Input | Algorithm | Output |
|---------------------------------|--|---|--|--|
| Chongchong <i>et al.</i> , 2011 | Hangzhou Bridge | Strain vibration, Distortion, and Cable tension | SVM + UL | Bridge structural damage degree |
| Li <i>et al.</i> , 2017 | NYC, Flushing 149th Bridge | Impact Echo (IE) signal collection | SVM | Damage classification of the deck |
| Alamdari <i>et al.</i> , 2016 | Sydney Harbour Bridge | Force, Acceleration, Time Histories | SVM | Crack detection |
| Li <i>et al.</i> , 2018 | Steel crane | Mode curvature change | SVM and Radial Basis Function | Damage degree identification |
| Oiwa <i>et al.</i> , 2017 | Wooden House | Vibration signals applied to a wooden brace | SVM, k-NN, and PCA | Degradation tracking, damage detection, and localization |
| Rankovic <i>et al.</i> , 2014 | Iron Gate 2 Serbia/Romania (Earth fill + Gravity Arch) Dam | H _{up} , H _{dn} , lag (H _{up}), lag (H _{dn}), OL | SVM | Tan_Displacement |
| Cheng and Zheng, 2013 | Gravity Dam | H _{up} , T _{air} , T _{conc} , Precip | SVM | Rad_Displacement, Uplift pressure |
| Jimenez <i>et al.</i> , 2019 | Wind Turbine | Guided ultrasonic waves, Guided Electrical signal simulating the effect of power, pitch angle, rotational speed, and wind speed | DT, DA, SVM, K-NN, EC, AR, PCA/NLARE, HNLPCA | Single-frequency and multi-frequency modes Ice thickness detection on blades |
| | Wind Turbine | Guided Electrical signal simulating the effect of power, pitch angle, rotational speed, and wind speed | ESD, kNN, LSVM, LDA, DT, AR, PCA / NCA | Dirt and mud detection on blades |
| Regan <i>et al.</i> , 2016 | Wind Turbine | Acoustic Based Signals | SVM, LR | Blades Damage prediction based on RMS, RSSQ, SD, Variance |
| (Zhang, Li, and Zhou, 2018) | Ocean University of china data Processing | Datastream coming from measured vibration data of the offshore turbines for data processing | SVM, Data clustering, Sym8 wavelet | Time-domain Feature SVM Classifier Test, Frequency-domain Feature SVM Classifier Test to detect the Global Damage prediction (Hole WT) |

2.4.3 Other Algorithms

Table 2.5 lists various algorithm applications in SHM. The Principal Component Analysis (PCA) algorithm was used for DD purposes in BHM, for instance in Japan's Hayakawa Truss Bridge (**Figure 2.8**), where data acquired from sensors installed on the bridge were deployed in the PCA algorithm combined with an Auto-Regressive (AR) model to detect damage (Unno *et al.*, 2019). Another application of this algorithm was in Taiwan's prestressed concrete Hanxi bridge, where data from single-channel deflection signals were used to detect deflection of concrete, shrinkage, and creep strains and prestress loss.

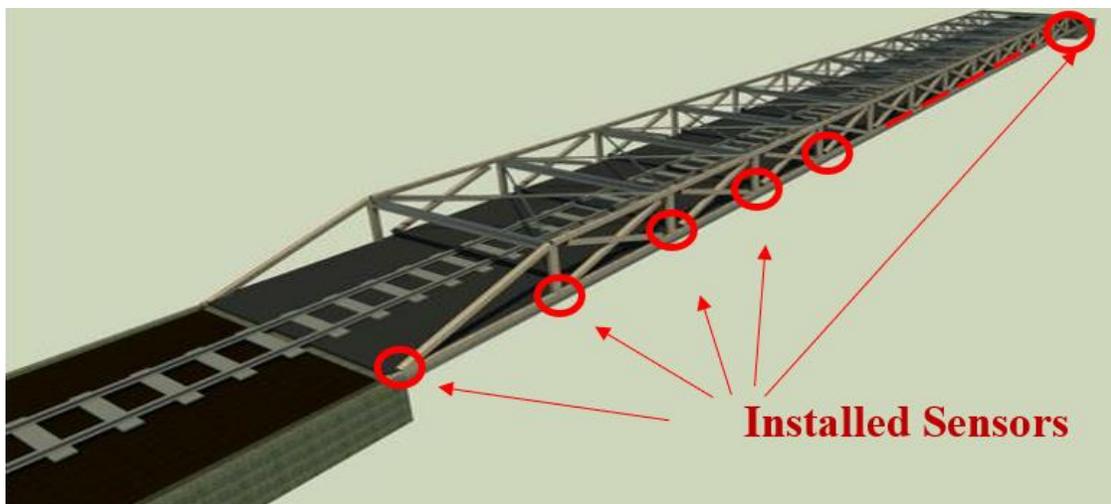


Figure 2.8: 3D Model of the Hayakawa Bridge, Japan.

One application of the Tree-structured Gaussian Process (TGP) algorithm was during the RM stage of BHM, where essential features related to the Tamar bridge in the UK were extracted, including its natural frequency, traffic loading applied to the bridge, wind direction and speed. Those features were introduced to the TGP algorithm to study the effects of wind conditions on the behavior of the main structural elements of the bridge. A second application was in Switzerland's Z24 Bridge, where modal parameters, air, and soil temperature, and soil humidity data were used to assess several parameters such as the settlement of the pier, landslide prediction, concrete spalling, concrete hinge failure, anchor head failure and the tendons rupture (Worden and Cross, 2018).

Table 2.5: Other ML Algorithms.

| Ref. | Structure | Input | Algorithm | Output |
|-----------------------------------|---|---|------------------------|---|
| Unno <i>et al.</i> , 2019 | Hayakawa Bridge (Hakone Tozan Railway), Japan | Vibration signals from sensors installed into the bridges | AR, PCA | Truss structures damage detection |
| Worden and Cross. 2018 | Bridge Z24, Switzerland | Modal parameters, air temperature, soil temperature, humidity | TGP | Pier settlement, Landslide, Concrete Spalling, Concrete hinge failure, Anchor head failure, Tendons rupture |
| | Bridge Tamar, UK | The natural frequency of the bridge, Traffic Loading, Wind direction and speed | TGP | Switching of the behavior due to variation of wind conditions |
| Rafiei and Adeli, 2018 | A prototype of a 38-story RC building structure | Ambient Vibration Response of the Structure collected by sensors | Deep Boltzmann Machine | Condition assessment, Structural Health Index |
| Cha and Buyukozturk, 2014 | Laboratory 3-dimensional steel structure | Modal Strain Energy | HMOO | Location and extent of induced multiple minor damages |
| Diez <i>et al.</i> , 2016 | Sydney Harbour, Australia | Vibration Signals of passing vehicles in joints | K-means clustering | Damaged Joints Detection and location |
| Salazar <i>et al.</i> , 2017 | Arch Dam, La Baells, Spain | H_up, T_amb, OL | BRT | Rad_Displacement |
| Kang <i>et al.</i> 2017, and 2019 | Gravity Dam, Fengman, China | Ux, T_amb, Δw | ELM | Rad_Displacement |
| Barahona <i>et al.</i> , 2017 | Wind Turbine | SCADA Vibration signals got from sensors installed on the WT | K-NN | Damage detection through the classification of WT operating regimes |
| Catbas and Malekzadeh, 2016 | Sunrise Movable Bridge in Ft. Lauderdale Florida, USA | Measured vibrations from Gearbox, rack, and pinion and Motor, Acoustic signals measured by microphones in the gearbox | CCA, RRA | Damage scenarios detection caused by leakage of enough oil in gearbox, |
| | | | | Bolt removal from rack and pinion |
| Ye <i>et al.</i> , 2018 | Prestressed Concrete Hanxi, Taiwan | Single Channel Deflection Signal | EEMD, PCA | The deflection of the girder: Concrete Shrinkage, Creep and Prestress loss |

A methodology to detect local and global health conditions of structural systems using ambient vibration response of structures collected by installed sensors was proposed (Rafiei and Adeli, 2018). Unsupervised deep Boltzmann machine (DBM) was combined with numerical methods such as wavelet and Fast Fourier transform to extract features from the frequency domain of the recorded signals and create a classification index for the local and global health of the structure using a probability density function. The algorithm was validated through a verification test case using actual experimental data obtained on a 1:20 scaled residential 42-story concrete building in Hong- Kong (**Figure 2.9**).

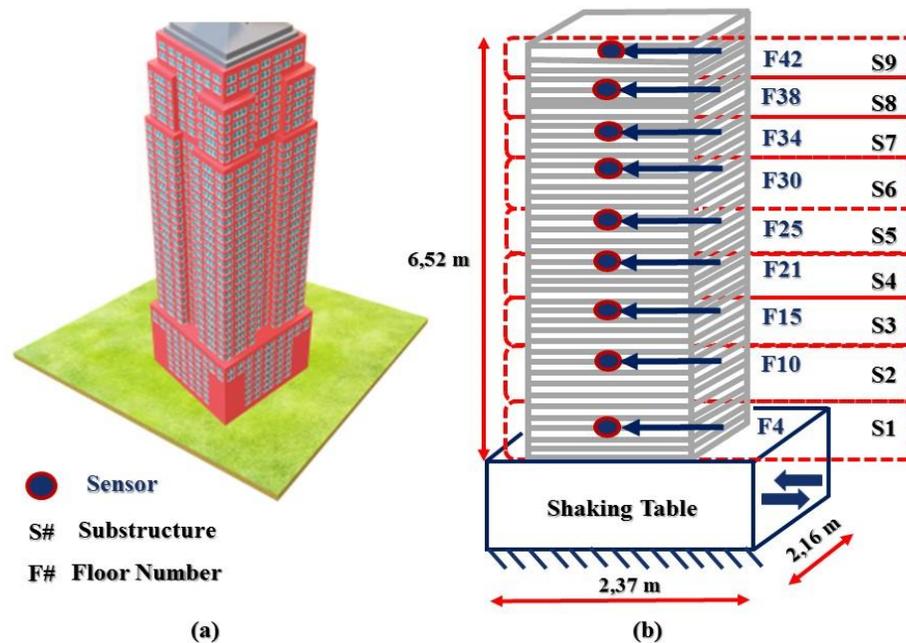


Figure 2.9: a) 3D model of the Hong-Kong 42 story High-rise Building. b) Scaled Prototype of the Substructures and Location of Sensors along with the Height of the Building.

A Hybrid Multi-Objective Optimization (HMOO) algorithm was proposed to detect damage by solving the inverse problem of limiting the change of modified modal strain energy in structural elements (Cha and Buyukozturk, 2014). A scaled model of the building was designed and then numerically modeled by Finite Element Analysis to assess the performance of the algorithm. The approach was compared to other traditional methods

using a single-objective Genetic Algorithm (GA). HMOO achieved better performance in detecting multiple minor damages, which had little effect on changing the modal properties of the structure. Moreover, the proposed method demonstrated the ability to mitigate difficulties of measuring rotational components of each mode shape using incomplete mode shapes that incorporated only global translational components.

The K-means clustering algorithm was also applied to detect and localize damage in joints of the Sydney Harbor Bridge, Australia (Diez *et al.*, 2016). Moreover, Bayesian Networks (BN) were deployed to rate the condition and structural reliability of the Albert railway bridge in Brisbane, Australia [46]. Another approach (Salazar *et al.*, 2017) used Boosted Regression Trees BRT combined with a 100-m finite element numerical model to detect anomalies in a dam (Rad_Disp) (**Figure 2.10**).

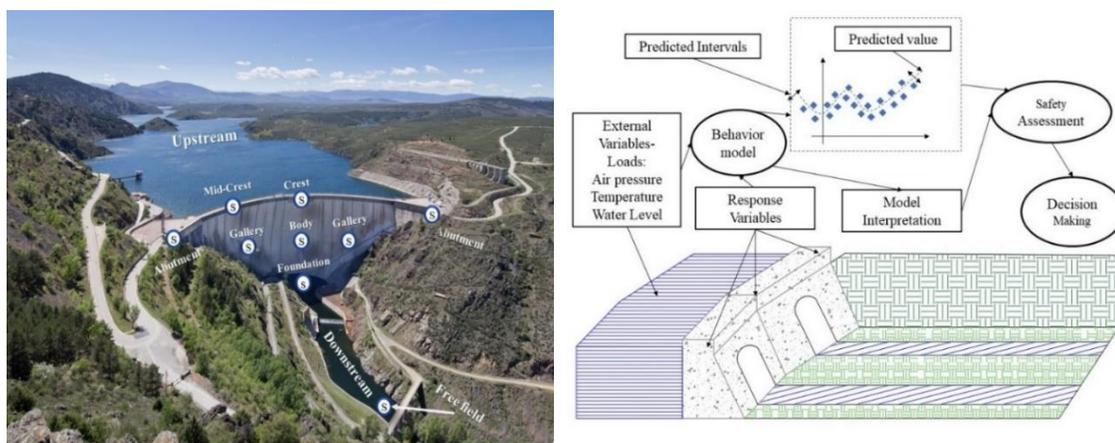


Figure 2.10: a) A disposition of the Installed Sensors in a Dam. b) Flow Diagram of DM Data Analysis.

This algorithm was effective compared to casual (only considering external variables, e.g., reservoir level) and non-casual models (including both internal and lagged variables as predictors). However, Kang *et al.*, 2017 and 2019) compared four sets of algorithms, namely BPNNs, Multiple Linear Regression (MLR), Step Wise Multiple Regression (SWMR) and Extreme Learning Machine (ELM) applied on a dataset obtained on the Fengman Dam in China and found that ELM was the most accurate algorithm. A technique called Pitch and Catch was used to detect ice thickness on blades using a combination of

Guided Ultrasonic Waves (GUW) and supervised ML algorithm. Several case studies of ice on the WTB surface have been used to test and validate the approach. NCA made the data needed to be well processed before running the algorithm, using four feature extraction methods, linear (Autoregressive (AR) and PCA) and nonlinear (nonlinear-AR exogenous and Hierarchical non-linear PCA), the feature selection. Twenty ML classifiers were used, including DT, DA, SVM, K-NN, and EC. The results were reasonably accurate and were verified in single frequency and multi-frequency modes (Jiménez *et al.*, 2019). A different study (Jiménez *et al.*, 2019) used the same technique with similar features to catch dirt and mud layers on WTB. The same supervised ML (pattern recognition) algorithm was used to classify signals based on the fault. Another application to detect damage on WTB was proposed in (Regan *et al.*, 2016) using an acoustic method based on Linear Regression (LR) and SVM algorithms combined with optimal feature selection to make accurate decisions. A laboratory-scale wind turbine was built, having an external microphone to monitor blade damage while being internally ensonified by wireless speakers.



Figure 2.11: Sensors for WTHM.

To detect integral health of wind turbines, (Zhang *et al.*, 2018) implemented a method to extract numeral characteristics and predict the health condition from data stream acquired from sensors, as illustrated in **Figure 2.11**. The SVM algorithm classifies the health

condition of the WTB online in both time and frequency domains based on a stream of data received from sensors installed on a WT in China. The algorithm proved the ability to detect online vibration and predict health conditions. Another application (Barahona *et al.*, 2017) proposed a method to classify the operating regimes from coarse resolution to Supervisory Control and Data Acquisition systems (SCADA) recorded by the turbine supervisory controller to finally classify damage of WT using K-NN algorithm with PCA to treat the data. Furthermore, a mix between nonlinear curve method and other ML algorithms (SVM with different kernel functions and BPNNs) has been set to detect scouring conditions along pipelines for thermometry-based Tunnel Health Monitoring (THM) (Zhao *et al.*, 2015). The SVM model with radial basis function was found to be the best classifier for scour monitoring, reaching 99.9% and 98.9% for accuracy for training and testing sets, respectively. Other references, such as (Catbas and Malekzadeh, 2016), measured the vibration of the gearbox, rack and pinion, and motor to detect damage in a movable bridge. Moreover, Ye *et al.* (2018) used a single-channel deflection signal for a prestressed concrete bridge employing PCA and Ensemble Empirical Modal Decomposition (EEMD) to detect the deflection of the girder, concrete shrinkage, creep, and prestress loss. Other ML algorithms and their corresponding uses are summarized in **Table 2.5**.

2.5 Analysis and Discussion

Tables 2.1-2.4 present a summary of different applications of ML and DL algorithms in the field of SHM. Based on the comprehensive review provided above, different applications, their advantages, and drawbacks, along with knowledge gaps research needs of the different algorithms of ML in SHM, have been identified and summarized.

PCA was primarily used to reduce the dimensions of data, which helps to reduce computational cost and to obtain higher accuracy in most cases. However, the problem of calculation time remains a drawback. PCA was used in (Datteo *et al.*, 2017) to model the vibration response of a stand in the Giuseppe-Meazza stadium, and **Figure 2.12** displays an outline of the installed sensors. The aim was to illustrate the state of the structure in 2D or 3D space principal directions and to interpret how this data processing considers the different effects of operational and environmental conditions. The results showed good

agreement with actual temperature and humidity values, and so is a good simulation for the behavior of the structure during major events like concerts and football matches.



Figure 2.12: Sensors Installed in Giuseppe Mazzei Stadium, Italy.

NNs can work with so-called “incomplete knowledge,” where it can produce output even with incomplete information after successful training. NNs perform very well with repetitive events, so it can learn and make decisions based on similar tasks already done (supervised learning). Another critical point is that NNs are tolerant to a certain point if one or more cells of the NN is corrupted, but this will not prevent it from having an output. Most applications in the open literature were in the field of DHM, because of the simplicity and accuracy of NN compared to traditional statistical and heuristic models. Despite their great success in some areas of research, NNs are now outdated in SHM applications. More advanced ML algorithms are being implemented to achieve a balance between the performance of the network and its computational time.

BPNNs can be easily distracted in the case of noisy data and can lead to erroneous results, including overfitting and drastic deterioration of the classification or regression task. However, BPNNs performed very well in bridge and building health monitoring, as mentioned in Section 4.1. One of the most significant advantages of BPNN is that it simplifies the network structure by removing the unnecessary weighted links that do not have a valuable effect on the trained network.

More recently, CNNs have proved their great success with DL tasks and especially computer vision-based applications. CNNs outperformed traditional neural networks on conventional image recognition, classification, and segmentation tasks. Another critical parameter of CNNs in image recognition, compared to conventional image processing techniques and other artificial neural networks, is that the features of the images are automatically extracted and do not require manual handling. Furthermore, CNNs are very efficient in pre-training tasks and can reduce the computational time and then save memory since the network does not have to be trained each time from scratch. Only the classifier must be trained based on the provided labels.

CNNs were first applied in SHM problems about five years ago. The primary application was aimed at detecting cracks as the first indicator of structural damage in sidewalks, asphalt pavements, concrete, and steel structures. Several sub-models employing CNNs are rapidly evolving, including Inception V2 and V3, ResNet 50 and 100, and many others. However, these kinds of networks need powerful computational configuration features (GPU) and massive data for training; otherwise, the network will overfit and lead to erroneous results.

SVM proved its effectiveness in binary classifications, training, building, and regression tasks. For instance, the SVM algorithm has one crucial feature called “L2 Regularization,” which is characterized by superior generalization capability. Another characteristic of SVM is that it performs very well in non-linear data from different sensors installed on structures. The processing of data has presented an obstacle for other kinds of neural networks, especially when there is a specific change in the data. On the contrary, SVM showed excellent stability since such change does not affect the hyperplane. However, the use of the SVM algorithm can be challenging since the filter, or the kernel needs to be appropriately chosen to handle non-linear data, and this can lead to generating too many support vectors, which will lead to more calculation time. Moreover, the data obtained from sensors need first to be scaled manually, which reduces the time to obtain classification and regression results effectively. SVM has been attributed to almost every kind of structure given its high accuracy when dealing with the problem of having a clear margin

of separation between classes (safe structure and damaged one), but its application is still dependent on the computation time, which is one of the most critical factors in AI tasks.

Other algorithms like TGP, HMOO, K-NN, K-means clustering, and ELM were proposed in 4.6. Those algorithms were used in several applications of SHM but did not achieve the popularity of NNs and SVM. For example, ELM was first proposed by in (Huang *et al.*, 2004, 2006, 2007, 2011) as a tool that is faster in the training phase, which may result in better interpolation but did not necessarily produce more precise and accurate results. For ML problems, more importance is assigned to the accuracy of the algorithm. Thus, ELM was not as credible in SHM applications.

In the present critical review, such methods have been divided into two main categories, namely vibration-based and image-based algorithms. The strengths and weaknesses of those algorithms were investigated and critically discussed. It has been found that more dedicated studies need to be performed concerning the following aspects:

Vibration-based algorithms need to concentrate more on wind-induced vibrations, especially for high-rise buildings, bridges, and towers. Moreover, other sophisticated algorithms can be applied in SHM of civil engineering structures since they have proved their applicability and high prediction accuracy in other fields, such as mechanical and aerospace engineering. These include Naïve Bayes (NB) classifier, Self-Organizing Maps (SOM) and k-means clustering (Nick *et al.*, 2015). However, the main issue with the applicability of these algorithms is the accuracy of the selection of the structure concerning the number of layers and the combined algorithms with those classifiers.

For image recognition tasks using CNNs, more research is needed to maintain a robust algorithm with high accuracy using small datasets and a smaller number of convolutional blocks that can affect the computation time and need for high computational resources. Furthermore, this algorithm should take care of the different distortions that can happen because of lighting conditions, shooting metric distance, angle of shooting, etc.

Most algorithms that are available in the open literature are supervised learning algorithms that need to be labeled manually. There is a need to implement unsupervised learning for

monitoring tasks using clustering to broaden the scope of applications of CNNs. Of the existing applications, about 95% have limited detection algorithms on the shallow scale of the distribution of cracks dealing with crack distribution, width, length, spalling, scaling, and efflorescence. More advanced studies go beyond that scope to determine whether the reinforcement is exposed, the steel rebars are corroded, etc. However, in order to make algorithms more robust and, therefore, more appealing to the industry, researchers need to relate these concepts not only to the diagnosis level but also to the damage mechanisms within concrete. For instance, several chemical mechanisms can occur underneath the concrete surface, while the exterior surface may appear integral and free of cracks and damage. Accordingly, further research is needed to cover the following aspects:

Relating crack initiation to concrete mixture design, curing conditions, mechanical and environmental conditions of the structure, such as the chemistry of the pore solution, mechanical loading, seismicity of the area, temperature, humidity, etc. Some phenomena that are dependent on those conditions include carbonation of the concrete cover, corrosion of steel reinforcement, freeze-thaw damage, sulfate attack, shrinkage strains and cracking, etc. While this is a significant undertaking, it could be done by combining available algorithms with experimental data of techniques such as infrared thermography, radar, impact-echo, and other ultrasonic techniques, half-cell potential and polarization scanning, etc. (Omar *et al.*, 2018). Some applications have related chemical, physical, and mechanical testing conditions to associated damage. A proof-of-concept evaluation of using CNNs was performed (Sanchez and Terra, 2019). The study aimed to identify damage features in images of concrete samples at a microscopic scale. This was based on a management protocol developed by Bérubé *et al.* (2005). Improved guidelines have then been proposed (in Sanchez *et al.*, 2016, 2017, and 2018) to optimize testing protocols and models and explore numerous distress processes in concrete, such as Alkali-Aggregate Reaction (AAR), Delayed Ettringite Formation (DEF), and cyclic Freezing and Thawing (FT). The developed approach was based on three phases. The first succeeded to predict seven different Damage Rating Indices (DRI) features, but with an average accuracy of only 64%, due to the limited number of the microscopic image dataset. The second aimed to use the same explicit DRI formula that an expert petrographer would apply based on crack counts. The third was aimed to use the refined ML algorithm for assessing other

damage mechanisms, such as external and internal sulfate attack, FT damage, and steel corrosion, to generate a comprehensive protocol that could be used to assess critical aging infrastructure. Ongoing research is being carried out to improve the accuracy of phase 1 by conducting more experiments and then providing additional training data. Phase 2 was still being processed. Phase 3 did not start yet, till phase 2 has been successfully implemented for AAR cases.

Relating the cause of cracks to structural conditions, for example by detecting mechanical loads causing the cracks, application of fracture mechanics with a possibility to predict the stress field around the crack (Bazant, 2019; Hillerborg *et al.*, 1976) and then assessing the remaining stresses that the structural element could resist in the short and long-term. This could be broadened by empowering the algorithm to propose solutions for the diagnosed problems based on available resources, such as the knowledge of experts, international codes, etc. Another evolving research item in this field is real-time concrete crack detection, which needs more consideration and more considerable efforts to transfer images to video rendering that could efficiently detect cracks on time.

2.6 Conclusions

There has been a rapid increase in the volume of research on applications of ML algorithms in the field of SHM. Such studies explore the essential benefits of ML, enhance its applicability and accuracy, and strive to reduce the associated computational effort. The application of ML algorithms to detect, assess, and possibly repair and rehabilitate damage in civil engineering structures is garnering increasing attention. We stand at the brink of a technological revolution where artificial intelligence could dominate what we do in SHM and the management of aging civil infrastructure assets. In this chapter, the leading techniques and algorithms that have been deployed for this purpose in the open literature have been critically surveyed, discussed, and analyzed. Detailed tables have been made to summarize the state-of-art and provide the reader with convenient access to the volume of work that has been conducted in this domain. The advantages and limitations of these techniques have been identified, and best practice recommendations for their use have been formulated. Knowledge gaps and future research needed have been outlined. This critical

review should better position engineers for decision making regarding the use of ML and DL algorithms in the domain of SHM.

2.7 Conflict of interest statement

The authors have no conflict of interest, whether implicit or explicit, related to this manuscript. The development of this work abides by the highest standards of ethics and collegial academic work.

2.8 References

- Agdas, D., Rice, J. A., Martinez, J. R. & Lasa, I. R. (2015), Comparison of Visual Inspection and Structural-Health Monitoring as Bridge Condition Assessment Methods, *Journal of Performance of Constructed Facilities*, **30**(3), 04015049.
- Agency, F. E. M. (2017), *Rapid Visual Screening of Buildings for Potential Seismic Hazards: A Handbook*, Government Printing Office.
- Alamdari, M. M., Khoa, N., Runcie, P., Li, J. & Mustapha, S. (2016), Characterization of Gradually Evolving Structural Deterioration in Jack Arch Bridges Using Support Vector Machine, *Maintenance, Monitoring, Safety, Risk and Resilience of Bridges and Bridge Networks-Proceedings of the 8th International Conference on Bridge Maintenance, Safety and Management, IABMAS 2016*.
- Ali, M. M. & Al-Kodmany, K. (2012), Tall Buildings and Urban Habitat of the 21st Century: A Global Perspective, *Buildings*, **2**(4), 384-423.
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van Esesn, B. C., Awwal, A. A. S. & Asari, V. K. (2018), The History Began from Alexnet: A Comprehensive Survey on Deep Learning Approaches, *arXiv preprint arXiv:1803.01164*.
- Amezquita-Sanchez, J. P. & Adeli, H. (2016), Signal Processing Techniques for Vibration-Based Health Monitoring of Smart Structures, *Archives of Computational Methods in Engineering*, **23**(1), 1-15.
- Avdelidis, N., Almond, D., Ibarra-Castanedo, C., Bendada, A., Kenny, S. & Maldague, X. (2006), Structural Integrity Assessment of Materials by Thermography, *Conf. Damage in Composite Materials CDCM, Stuttgart, Germany*, Citeseer.
- Balageas, D., Fritzen, C.-P. & Güemes, A. (2010), *Structural Health Monitoring*, John Wiley & Sons.

- Barahona, B., Hoelzl, C. & Chatzi, E. (2017), Applying Design Knowledge and Machine Learning to Scada Data for Classification of Wind Turbine Operating Regimes, *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, pp. 1-8.
- Bazant, Z. P. (2019), *Fracture and Size Effect in Concrete and Other Quasibrittle Materials*, Routledge.
- Beckman, G. H., Polyzois, D. & Cha, Y.-J. (2019), Deep Learning-Based Automatic Volumetric Damage Quantification Using Depth Camera, *Automation in Construction*, **99**, 114-124.
- Bérubé, M.-A., Smaoui, N., Fournier, B., Bissonnette, B. & Durand, B. (2005), Evaluation of the Expansion Attained to Date by Concrete Affected by Alkali-Silica Reaction. Part Iii: Application to Existing Structures, *Canadian Journal of Civil Engineering*, **32**(3), 463-479.
- Brooks, R. A. (1991), Intelligence without Representation, *Artificial intelligence*, **47**(1-3), 139-159.
- Brooks, R. A. (1991), New Approaches to Robotics, *Science*, **253**(5025), 1227-1232.
- Brown, C. A. & Graham, W. J. (1988), Assessing the Threat to Life from Dam Failure 1, *JAWRA Journal of the American Water Resources Association*, **24**(6), 1303-1309.
- Brownjohn, J. M. (2006), Structural Health Monitoring of Civil Infrastructure, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **365**(1851), 589-622.
- Bulut, A., Singh, A. K., Shin, P., Fountain, T., Jasso, H., Yan, L. & Elgamal, A. (2005), Real-Time Nondestructive Structural Health Monitoring Using Support Vector Machines and Wavelets, *Advanced Sensor Technologies for Nondestructive Evaluation and Structural Health Monitoring*, International Society for Optics and Photonics, pp. 180-189.
- Burges, C. J. (1998). A Tutorial on Support Vector Machines for Pattern Recognition, *Data mining and knowledge discovery*, **2**(2), 121-167.
- Catbas, F. N. & Malekzadeh, M. (2016), A Machine Learning-Based Algorithm for Processing Massive Data Collected from the Mechanical Components of Movable Bridges, *Automation in Construction*, **72**, 269-278.
- Cha, Y.-J. & Buyukozturk, O. (2014), Modal Strain Energy-Based Damage Detection Using Multi-Objective Optimization, *Structural Health Monitoring, Volume 5*, Springer, pp. 125-133.

- Chang, C.-M., Lin, T.-K. & Chang, C.-W. (2018) Applications of Neural Network Models for Structural Health Monitoring Based on Derived Modal Properties, *Measurement*, **129**, 457-470.
- Chen, F.-C. & Jahanshahi, M. R. (2017), Nb-Cnn: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion, *IEEE Transactions on Industrial Electronics*, **65**(5), 4392-4400.
- Chen, H.-P. & Ni, Y.-Q. (2018), *Structural Health Monitoring of Large Civil Engineering Structures*, Wiley Online Library.
- Cheng, L. & Zheng, D. (2013), Two Online Dam Safety Monitoring Models Based on the Process of Extracting Environmental Effect, *Advances in Engineering Software*, **57**, 48-56.
- Chongchong, Y., Jingyan, W., Li, T. & Xuyan, T. (2011), A Bridge Structural Health Data Analysis Model Based on Semi-Supervised Learning, *2011 IEEE International Conference on Automation and Logistics (ICAL)*, IEEE, pp. 30-34.
- Ciang, C. C., Lee, J.-R. & Bang, H.-J. (2008), Structural Health Monitoring for a Wind Turbine System: A Review of Damage Detection Methods, *Measurement science, and technology*, **19**(12), 122001.
- Dams, I. C. o. L. (2012), Dam Surveillance Guide, *ICOLD*.
- Datteo, A., Lucà, F. & Busca, G. (2017), Statistical Pattern Recognition Approach for Long-Time Monitoring of the G. Meazza Stadium by Means of Ar Models and Pca, *Engineering Structures*, **153**, 317-333.
- de Almeida Cardoso, R., Cury, A. & Barbosa, F. (2019), Automated Real-Time Damage Detection Strategy Using Raw Dynamic Measurements, *Engineering Structures*, **196**, 109364.
- Demirkaya, S. & Balcilar, M. (2012), The Contribution of Soft Computing Techniques for the Interpretation of Dam Deformation, *Proceedings of the FIG working week, Rome, Italy*.
- Dhital, D. & Lee, J.-R. (2012), A Fully Non-Contact Ultrasonic Propagation Imaging System for Closed Surface Crack Evaluation, *Experimental mechanics*, **52**(8), 1111-1122.
- Diez, A., Khoa, N. L. D., Alamdari, M. M., Wang, Y., Chen, F. & Runcie, P. (2016), A Clustering Approach for Structural Health Monitoring on Bridges, *Journal of Civil Structural Health Monitoring*, **6**(3), 429-445.

- Dorafshan, S., Thomas, R. J. & Maguire, M. (2018), Comparison of Deep Convolutional Neural Networks and Edge Detectors for Image-Based Crack Detection in Concrete, *Construction, and Building Materials*, **186**, 1031-1045.
- Dudani, S. A. (1976), The Distance-Weighted K-Nearest-Neighbor Rule, *IEEE Transactions on Systems, Man, and Cybernetics*(4), 325-327.
- Dung, C. V., Sekiya, H., Hirano, S., Okatani, T. & Miki, C. (2019), A Vision-Based Method for Crack Detection in Gusset Plate Welded Joints of Steel Bridges Using Deep Convolutional Neural Networks, *Automation in Construction*, **102**, 217-229.
- Fan, J., Yuan, Y. & Cao, X. (2015), Developing Situation and Research Advances of Structural Damage Detection Using Bp Network, *2015 4th National Conference on Electrical, Electronics and Computer Engineering*, Atlantis Press.
- Farrar, C. R. & Worden, K. (2012), *Structural Health Monitoring.: A Machine Learning Perspective*, John Wiley & Sons.
- Fawzi, A., Samulowitz, H., Turaga, D. & Frossard, P. (2016), Adaptive Data Augmentation for Image Classification, *2016 IEEE International Conference on Image Processing (ICIP)*, Ieee, pp. 3688-3692.
- Fisher, W. D., Camp, T. K. & Krzhizhanovskaya, V. V. (2017), Anomaly Detection in Earth Dam and Levee Passive Seismic Data Using Support Vector Machines and Automatic Feature Selection, *Journal of Computational Science*, **20**, 143-153.
- Frangopol, D. M. & Soliman, M. (2016), Life-Cycle of Structural Systems: Recent Achievements and Future Directions, *Structure and infrastructure engineering*, **12**(1), 1-20.
- Fujita, Y. & Hamamoto, Y. (2011), A Robust Automatic Crack Detection Method from Noisy Concrete Surfaces, *Machine Vision and Applications*, **22**(2), 245-254.
- Gao, Y. & Mosalam, K. M. (2018), Deep Transfer Learning for Image-Based Structural Damage Recognition, *Computer-Aided Civil and Infrastructure Engineering*, **33**(9), 748-768.
- González, M. P. & Zapico, J. L. (2008), Seismic Damage Identification in Buildings Using Neural Networks and Modal Data, *Computers & structures*, **86**(3-5), 416-426.
- Gros, X. E. (1995), An Eddy Current Approach to the Detection of Damage Caused by Low-Energy Impacts on Carbon Fibre Reinforced Materials, *Materials & Design*, **16**(3), 167-173.
- Gunn, R. M. (2015), Proceedings of the 13th Icold, International Benchmark Workshop on, the Numerical Analysis of Dams

- Hadjipaschalis, I., Poullikkas, A. & Efthimiou, V. (2009), Overview of Current and Future Energy Storage Technologies for Electric Power Applications, *Renewable and sustainable energy reviews*, **13**(6-7), 1513-1522.
- Hao, H., Zhang, W., Li, J. & Ma, H. (2018), Bridge Condition Assessment under Moving Loads Using Multi-Sensor Measurements and Vibration Phase Technology, *Engineering Asset Management 2016*, Springer, pp. 73-84.
- Hasni, H., Alavi, A. H., Jiao, P. & Lajnef, N. (2017), Detection of Fatigue Cracking in Steel Bridge Girders: A Support Vector Machine Approach, *Archives of Civil and Mechanical Engineering*, **17**(3), 609-622.
- Hecht-Nielsen, R. (1992), Theory of the Backpropagation Neural Network, *Neural Networks for Perception*, Elsevier, pp. 65-93.
- Hillerborg, A., Modéer, M. & Petersson, P.-E. (1976), Analysis of Crack Formation and Crack Growth in Concrete by Means of Fracture Mechanics and Finite Elements, *Cement and concrete research*, **6**(6), 773-781.
- Huang, G.-B. & Chen, L. (2007), Convex Incremental Extreme Learning Machine, *Neurocomputing*, **70**(16-18), 3056-3062.
- Huang, G.-B., Zhou, H., Ding, X. & Zhang, R. (2011), Extreme Learning Machine for Regression and Multiclass Classification, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **42**(2), 513-529.
- Huang, G.-B., Zhu, Q.-Y. & Siew, C.-K. (2004), Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks, *Neural networks*, **2**, 985-990.
- Huang, G.-B., Zhu, Q.-Y. & Siew, C.-K. (2006), Extreme Learning Machine: Theory and Applications, *Neurocomputing*, **70**(1-3), 489-501.
- Hüthwohl, P., Lu, R. & Brilakis, I. (2019), Multi-Classifer for Reinforced Concrete Bridge Defects, *Automation in Construction*, **105**, 102824.
- Jiménez, A. A., Márquez, F. P. G., Moraleda, V. B. & Muñoz, C. Q. G. (2019), Linear and Nonlinear Features and Machine Learning for Wind Turbine Blade Ice Detection and Diagnosis, *Renewable energy*, **132**, 1034-1048.
- Jiménez, A. A., Muñoz, C. Q. G. & Márquez, F. P. G. (2019), Dirt and Mud Detection and Diagnosis on a Wind Turbine Blade Employing Guided Waves and Supervised Learning Classifiers, *Reliability Engineering & System Safety*, **184**, 2-12.
- Jolliffe, I. (2011), Principal Component Analysis, in Lovric, M. (ed.), *International Encyclopedia of Statistical Science*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1094-1096.

- Jordan, M. I. & Mitchell, T. M. (2015), Machine Learning: Trends, Perspectives, and Prospects, *Science*, **349**(6245), 255-260.
- Kang, F., Li, J. & Dai, J. (2019), Prediction of Long-Term Temperature Effect in Structural Health Monitoring of Concrete Dams Using Support Vector Machines with Jaya Optimizer and Salp Swarm Algorithms, *Advances in Engineering Software*, **131**, 60-76.
- Kang, F., Liu, J., Li, J. & Li, S. (2017), Concrete Dam Deformation Prediction Model for Health Monitoring Based on Extreme Learning Machine, *Structural Control and Health Monitoring*, **24**(10), e1997.
- Kao, C. Y. & Loh, C. H. (2013), Monitoring of Long-Term Static Deformation Data of Fei-Tsui Arch Dam Using Artificial Neural Network-Based Approaches, *Structural Control and Health Monitoring*, **20**(3), 282-303.
- Karballaezadeh, N., Mohammadzadeh S, D., Shamsirband, S., Hajikhodaverdikhan, P., Mosavi, A. & Chau, K.-w. (2019), Prediction of Remaining Service Life of Pavement Using an Optimized Support Vector Machine (Case Study of Semnan–Firuzkuh Road), *Engineering Applications of Computational Fluid Mechanics*, **13**(1), 188-198.
- Kerh, T. & Ting, S. (2005), Neural Network Estimation of Ground Peak Acceleration at Stations Along with Taiwan High-Speed Rail System, *Engineering Applications of Artificial Intelligence*, **18**(7), 857-866.
- Kim, B. & Cho, S. (2018), Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique, *Sensors*, **18**(10), 3452.
- Kim, D., Liu, M., Lee, S. & Kamat, V. R. (2019), Remote Proximity Monitoring between Mobile Construction Resources Using camera-mounted UAVs, *Automation in Construction*, **99**, 168-182.
- 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.
- Kim, H., Lee, J., Ahn, E., Cho, S., Shin, M. & Sim, S.-H. (2017), Concrete Crack Identification Using a Uav Incorporating Hybrid Image Processing, *Sensors*, **17**(9), 2052.
- Kim, I.-H., Jeon, H., Baek, S.-C., Hong, W.-H. & Jung, H.-J. (2018), Application of Crack Identification Techniques for an Aging Concrete Bridge Inspection Using an Unmanned Aerial Vehicle, *Sensors*, **18**(6), 1881.
- LeCun, Y., Bengio, Y. & Hinton, G. (2015), Deep Learning, *nature*, **521**(7553), 436.

- Lee, D., Kim, J. & Lee, D. (2019), Robust Concrete Crack Detection Using Deep Learning-Based Semantic Segmentation, *International Journal of Aeronautical and Space Sciences*, **20**(1), 287-299.
- Li, B., Ushiroda, K., Yang, L., Song, Q. & Xiao, J. (2017), Wall-Climbing Robot for Non-Destructive Evaluation Using Impact-Echo and Metric Learning Svm, *International Journal of Intelligent Robotics and Applications*, **1**(3), 255-270.
- Li, S., Zhao, X. & Zhou, G. (2019), Automatic Pixel-Level Multiple Damage Detection of Concrete Structure Using Fully Convolutional Network, *Computer-Aided Civil and Infrastructure Engineering*, **34**(7), 616-634.
- Li, X., Xi, H., Zhou, C., Gu, W. & Gao, T. (2018), Damage Degree Identification of Crane Girder Based on the Support Vector Machine, *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, IEEE, pp. 920-924.
- Liang, X. (2019), Image-Based Post-Disaster Inspection of Reinforced Concrete Bridge Systems Using Deep Learning with Bayesian Optimization, *Computer-Aided Civil and Infrastructure Engineering*, **34**(5), 415-430.
- Limongelli, M. P. (2019), *Seismic Structural Health Monitoring: From Theory to Successful Applications*, Springer.
- Liu, Y., Yao, J., Lu, X., Xie, R. & Li, L. (2019), Deepcrack: A Deep Hierarchical Feature Learning Architecture for Crack Segmentation, *Neurocomputing*, **338**, 139-153.
- Maeda, H., Sekimoto, Y., Seto, T., Kashiya, T. & Omata, H. (2018), Road Damage Detection Using Deep Neural Networks with Images Captured through a Smartphone, *arXiv preprint arXiv:1801.09454*.
- Manuello, A., Niccolini, G. & Carpinteri, A. (2019), Ae Monitoring of a Concrete Arch Road Tunnel: Damage Evolution and Localization, *Engineering Fracture Mechanics*, **210**, 279-287.
- Martarelli, M., Revel, G. & Santolini, C. (2001), Automated Modal Analysis by Scanning Laser Vibrometry: Problems and Uncertainties Associated with the Scanning System Calibration, *Mechanical Systems and Signal Processing*, **15**(3), 581-601.
- Mata, J. (2011), Interpretation of Concrete Dam Behaviour with Artificial Neural Network and Multiple Linear Regression Models, *Engineering Structures*, **33**(3), 903-910.
- Matsuzaki, R. & Todoroki, A. (2006), Wireless Detection of Internal Delamination Cracks in Cfrp Laminates Using Oscillating Frequency Changes, *Composites science and technology*, **66**(3-4), 407-416.

- Mehrjoo, M., Khaji, N., Moharrami, H. & Bahreininejad, A. (2008), Damage Detection of Truss Bridge Joints Using Artificial Neural Networks, *Expert Systems with Applications*, **35**(3), 1122-1131.
- Murao, S., Nomura, Y., Furuta, H. & Kim, C.-W. (2019), Concrete Crack Detection Using Uav and Deep Learning.
- Ni, F., Zhang, J. & Chen, Z. (2019), Zernike-Moment Measurement of Thin-Crack Width in Images Enabled by Dual-Scale Deep Learning, *Computer-Aided Civil and Infrastructure Engineering*, **34**(5), 367-384.
- Nick, W., Asamene, K., Bullock, G., Esterline, A. & Sundaresan, M. (2015), A Study of Machine Learning Techniques for Detecting and Classifying Structural Damage, *International Journal of Machine Learning and Computing*, **5**(4), 313.
- Nourani, V. & Babakhani, A. (2012), Integration of Artificial Neural Networks with Radial Basis Function Interpolation in Earthfill Dam Seepage Modeling, *Journal of Computing in Civil Engineering*, **27**(2), 183-195.
- Ochieng, F. X., Hancock, C. M., Roberts, G. W., Le Kernec, J. & Tang, X. (2018), Novel Non-Contact Deformation Health Monitoring of Towers and Rotating Composite Based Wind Turbine Blades Using Interferometric Ground-Based Radar.
- Oiwa, R., Ito, T. & Kawahara, T. (2017), Timber Health Monitoring Using Piezoelectric Sensor and Machine Learning, *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, IEEE, pp. 123-128.
- Oliveira, S. & Alegre, A. (2019), Seismic and Structural Health Monitoring of Dams in Portugal, *Seismic Structural Health Monitoring*, Springer, pp. 87-113.
- Omar, T. & Nehdi, M. L. (2016), Mat-713: Evaluation of Ndt Techniques for Concrete Bridge Decks Using Fuzzy Analytical Hierarchy Process.
- Omar, T., Nehdi, M. L. & Zayed, T. (2018), Infrared Thermography Model for Automated Detection of Delamination in Rc Bridge Decks, *Construction and Building Materials*, **168**, 313-327.
- Pao, Y. (1989) Adaptive Pattern Recognition and Neural Networks.
- Peng, J., Zhang, S., Peng, D. & Liang, K. (2017), Application of Machine Learning Method in Bridge Health Monitoring, *2017, 2nd International Conference on Reliability Systems Engineering (ICRSE)*, IEEE, pp. 1-7.
- Popovici, A., Ilinca, C. & Ayvaz, T. (2013), The Performance of the Neural Networks to Model Some Response Parameters of a Buttress Dam to Environment Actions, *Proceedings of the 9th ICOLD European club symposium, Venice, Italy*.

- Protopapadakis, E., Voulodimos, A., Doulamis, A., Doulamis, N. & Stathaki, T. (2019), Automatic Crack Detection for Tunnel Inspection Using Deep Learning and Heuristic Image Post-Processing, *Applied Intelligence*, **49**(7), 2793-2806.
- Rafiei, M. H. & Adeli, H. (2017), A Novel Machine Learning-Based Algorithm to Detect Damage in High-Rise Building Structures, *The Structural Design of Tall and Special Buildings*, **26**(18), e1400.
- Rafiei, M. H. & Adeli, H. (2018), A Novel Unsupervised Deep Learning Model for Global and Local Health Condition Assessment of Structures, *Engineering Structures*, **156**, 598-607.
- Ranković, V., Grujović, N., Divac, D. & Milivojević, N. (2014), Development of Support Vector Regression Identification Model for Prediction of Dam Structural Behaviour, *Structural Safety*, **48**, 33-39.
- Ranković, V., Novaković, A., Grujović, N., Divac, D. & Milivojević, N. (2014), Predicting Piezometric Water Level in Dams Via Artificial Neural Networks, *Neural Computing, and Applications*, **24**(5), 1115-1121.
- Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016), You Only Look Once: Unified, Real-Time Object Detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788.
- Regan, T., Canturk, R., Slavkovsky, E., Niezrecki, C. & Inalpolat, M. (2016), Wind Turbine Blade Damage Detection Using Various Machine Learning Algorithms, *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, pp. V008T010A040-V008T010A040.
- Riquelme, F., Fraile, J., Santillán, D., Morán, R. & Toledo, M. (2011), Application of Artificial Neural Network Models to Determine Movements in an Arch Dam, *Proceedings of the 2nd international congress on dam maintenance and rehabilitation. Zaragoza, Spain*, pp. 117-123.
- Salazar, F., Toledo, M., Oñate, E. & Morán, R. (2015). An Empirical Comparison of Machine Learning Techniques for Dam Behaviour Modelling, *Structural Safety*, **56**, 9-17.
- Salazar, F., Toledo, M. Á., González, J. M. & Oñate, E. (2017), Early Detection of Anomalies in Dam Performance: A Methodology Based on Boosted Regression Trees, *Structural Control and Health Monitoring*, **24**(11), e2012.
- Salloum, S., Huang, J. Z. & He, Y. (2019), Exploring and Cleaning Big Data with Random Sample Data Blocks, *Journal of Big Data*, **6**(1), 45.

- Sanchez, L., Drimalas, T., Fournier, B., Mitchell, D. & Bastien, J. (2018), Comprehensive Damage Assessment in Concrete Affected by Different Internal Swelling Reaction (Isr) Mechanisms, *Cement and concrete research*, **107**, 284-303.
- Sanchez, L., Fournier, B., Jolin, M., Bedoya, M., Bastien, J. & Duchesne, J. (2016), Use of Damage Rating Index to Quantify Alkali-Silica Reaction Damage in Concrete: Fine Versus Coarse Aggregate, *ACI Materials Journal*, **113**(4).
- Sanchez, L., Fournier, B., Jolin, M., Mitchell, D. & Bastien, J. (2017), Overall Assessment of Alkali-Aggregate Reaction (Aar) in Concretes Presenting Different Strengths and Incorporating a Wide Range of Reactive Aggregate Types and Natures, *Cement and concrete research*, **93**, 17-31.
- Sanchez, L. F. & Terra, M. (2019). Using Machine Learning for Condition Assessment of Concrete Infrastructure, *Concrete International*, **41**(11), 35-39.
- Santillán, D., Fraile-Ardanuy, J. & Toledo, M. (2014), Prediction of Gauge Readings of Filtration in Arch Dams Using Artificial Neural Networks, *Tecnología y Ciencias del Agua*, **5**(3), 81-96.
- Shan, B., Zheng, S. & Ou, J. (2016), A Stereovision-Based Crack Width Detection Approach for Concrete Surface Assessment, *KSCE Journal of Civil Engineering*, **20**(2), 803-812.
- Simon, A., Royer, M., Mauris, F. & Fabre, J. (2013), Analysis and Interpretation of Dam Measurements Using Artificial Neural Networks, *Proceedings of the 9th ICOLD European club symposium, Venice, Italy*.
- Simonyan, K. & Zisserman, A. (2014), Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv preprint arXiv:1409.1556*.
- Siringoringo, D. M. & Fujino, Y. (2006), Experimental Study of Laser Doppler Vibrometer and Ambient Vibration for Vibration-Based Damage Detection, *Engineering Structures*, **28**(13), 1803-1815.
- Smarsly, K., Dragos, K. & Wiggenbrock, J. (2016), Machine Learning Techniques for Structural Health Monitoring, *Proceedings of the 8th European Workshop on Structural Health Monitoring (EWSHM 2016), Bilbao, Spain*, pp. 5-8.
- Snell, J., Swersky, K. & Zemel, R. (2017), Prototypical Networks for Few-Shot Learning, *Advances in Neural Information Processing Systems*, pp. 4077-4087.
- Sørensen, B. F., Lading, L., Sendrup, P., McGugan, M., Debel, C. P., Kristensen, O. J., Larsen, G. C., Hansen, A. M., Rheinländer, J. & Rusborg, J. (2002), Fundamentals for Remote Structural Health Monitoring of Wind Turbine Blades-a Preproject.

- Soyoz, S. & Feng, M. Q. (2009), Long-Term Monitoring and Identification of Bridge Structural Parameters, *Computer-Aided Civil and Infrastructure Engineering*, **24**(2), 82-92.
- Su, H., Wen, Z., Sun, X. & Yang, M. (2015), Time-Varying Identification Model for Dam Behavior Considering Structural Reinforcement, *Structural Safety*, **57**, 1-7.
- Sutherland, H., Beattie, A., Hansche, B., Musial, W., Allread, J., Johnson, J. & Summers, M. (1994), The Application of Non-Destructive Techniques to the Testing of a Wind Turbine Blade, Sandia National Labs., Albuquerque, NM (United States).
- Takeda, N. (2002), Characterization of Microscopic Damage in Composite Laminates and Real-Time Monitoring by Embedded Optical Fiber Sensors, *International Journal of Fatigue*, **24**(2-4), 281-289.
- Unno, K., Mikami, A. & Shimizu, M. (2019), Damage Detection of Truss Structures by Applying Machine Learning Algorithms, *International Journal*, **16**(54), 62-67.
- Verijenko, B. & Verijenko, V. (2005), A New Structural Health Monitoring System for Composite Laminates, *Composite structures*, **71**(3-4), 315-319.
- Wang, K., Zhang, A., Li, J. Q., Fei, Y., Chen, C. & Li, B. (2017), Deep Learning for Asphalt Pavement Cracking Recognition Using Convolutional Neural Network, *Proc. Int. Conf. Airfield Highway Pavements*, pp. 166-177.
- Wang, N., Zhao, X., Zhao, P., Zhang, Y., Zou, Z. & Ou, J. (2019), Automatic Damage Detection of Historic Masonry Buildings Based on Mobile Deep Learning, *Automation in Construction*, **103**, 53-66.
- Weiss, G. (1999), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT press.
- Worden, K. & Cross, E. (2018), On Switching Response Surface Models, with Applications to the Structural Health Monitoring of Bridges, *Mechanical Systems and Signal Processing*, **98**, 139-156.
- Wu, X. (2004), Data Mining: Artificial Intelligence in Data Analysis, *Proceedings. IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004.(IAT 2004)*. IEEE, pp. 7.
- Wu, Z., Shen, C. & Van Den Hengel, A. (2019), Wider or Deeper: Revisiting the Resnet Model for Visual Recognition, *Pattern Recognition*, **90**, 119-133.
- Xu, K., Deng, Q., Cai, L., Ho, S. & Song, G. (2018), Damage Detection of a Concrete Column Subject to Blast Loads Using Embedded Piezoceramic Transducers, *Sensors*, **18**(5), 1377.

- Yang, J., Zhou, J. & Wang, F. (2008), A Study on the Application of Ga-Bp Neural Network in the Bridge Reliability Assessment, *2008 International Conference on Computational Intelligence and Security*, IEEE, pp. 540-545.
- Yang, X., Li, H., Yu, Y., Luo, X., Huang, T. & Yang, X. (2018), Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network, *Computer-Aided Civil and Infrastructure Engineering*, **33**(12), 1090-1109.
- Ye, X., Chen, X., Lei, Y., Fan, J. & Mei, L. (2018), An Integrated Machine Learning Algorithm for Separating the Long-Term Deflection Data of Prestressed Concrete Bridges, *Sensors*, **18**(11), 4070.
- Yuansong, L., Xinping, L. & Aiping, Y. (2007) The Prediction Method of Long-Span Cable-Stayed Bridge Construction Control Based on Bp Neural Network, *Proceedings of the 9th WSEAS international conference on Mathematical and computational methods in science and engineering*, World Scientific and Engineering Academy and Society (WSEAS), pp. 217-222.
- Zajec, B., Bajt Leban, M., Kosec, T., Kuhar, V., Legat, A., Lenart, S., Fifer Bizjak, K. & Gavin, K. (2018), Corrosion Monitoring of Steel Structure Coating Degradation, *Tehnički vjesnik*, **25**(5), 1348-1355.
- Zhang, A., Li, M. & Zhou, L. (2018), Structural Health Monitoring of Offshore Wind Turbine Based on Online Data-Driven Support Vector Machine, *2018 IEEE 7th Data-Driven Control and Learning Systems Conference (DDCLS)*, IEEE, pp. 990-995.
- Zhang, Y., Sun, X., Loh, K. J., Su, W., Xue, Z. & Zhao, X. (2019), Autonomous Bolt Loosening Detection Using Deep Learning, *Structural Health Monitoring*, 1475921719837509.
- Zhao, H., Zhang, X., Ji, L., Hu, H. & Li, Q. (2014), Quantitative Structure–Activity Relationship Model for Amino Acids as Corrosion Inhibitors Based on the Support Vector Machine and Molecular Design, *Corrosion Science*, **83**, 261-271.
- Zhao, X., Li, W., Zhou, L., Song, G., Ba, Q., Ho, S. C. M. & Ou, J. (2015), Application of Support Vector Machine for Pattern Classification of Active Thermometry-Based Pipeline Scour Monitoring, *Structural Control and Health Monitoring*, **22**(6), 903-918.

Chapter 3

3 Classification and Quantification of Cracks in Concrete Structures using 2D CNN Image-Based Techniques

Visual inspection has been the most widely used technique for monitoring concrete structures in service. Inspectors visually evaluate defects based on experience, skill, and engineering judgment. However, this process is subjective, laborious, time-consuming, and hampered by demanding access to numerous parts of complex structures. Accordingly, the present study proposes a nearly automated inspection model based on image processing and DL for detecting defects in typically inaccessible areas of concrete structures. The type of structural damage and its severity are identified based on the allowed range of concrete crack width for different structures, including buildings and bridges, based on different international standards and codes. The proposed method can deploy unmanned aerial vehicle image acquisition to offer a nearly automated inspection platform for the colossal backlog of aging concrete structures.

3.1 Introduction

A colossal backlog of aging civil infrastructure assets that need an inspection, repair, and rehabilitation has been generated in many countries around the world. For instance, 40% of the 570,000 bridges in the USA were classified as deficient, requiring rehabilitation or replacement according to the FHWA criteria, with an estimated cost of 50 billion dollars (Nowak *et al.*, 2012). Civil structures and infrastructures such as bridges, tunnels, buildings, dams, and roads are prone to damage due to various mechanisms related to mechanical loading, chemical processes, and environmental actions (e.g., Hong *et al.*, 2000). Hence, numerous structural SHM techniques have been proposed for detecting, locating, and monitoring such damage.

For instance, a study used embedded piezoceramic transducers for damage detection of a 6.1 m long reinforced concrete bridge bent cap (Song *et al.*, 2007). Their results showed that using piezoceramic transducers along with a damage index based on wavelet packet analysis was useful in identifying the occurrence and severity of cracks. In another approach, the dynamic properties of an elevated water reservoir were monitored via an

optical biaxial accelerometer attached to the structure (Antunes *et al.*, 2012). Their results showed that the dynamic properties of the high-water reservoir in the longitudinal and transverse directions could be determined using such an accelerometer. Furthermore, another study proposed a system consisting of a mobile robot and a crack detection system for inspecting and measuring cracks in concrete structures (Yu *et al.*, 2007). The resulting data was considered objective and could be used in evaluating safety. (Akbar *et al.*, 2019) investigated an unmanned aerial vehicle (UAV) based autonomous SHM system. In their study, images of the structural site captured by the UAV were stitched together to form a complete view of the structure using a speeded-up robust (SURF) based feature detection algorithm. As defined by (Koch *et al.*, 2015), Image stitching is a standard method to combine and visualize a collection of images. Another computer vision-based defect detection and condition assessment tool for civil infrastructure has also been proposed in recent years. In order to increase the level of automation on concrete infrastructure inspection using UAV, a more contemporary image processing tool in the field of image pattern recognition called CNN has received considerable attention. CNN is a powerful image processing tool that uses DL to deal with three main tasks: (a) object classification, which classifies the type of the object, (b) object detection, which determines the location of the object using a rectangular bounding box, and (c) image segmentation, which divides the image into groups of pixels according to a specific pattern (Simonyan *et al.*, 2014; Pal *et al.*, 1993; Simonyan *et al.*, 2014). For example, (Kim *et al.*, 2019) presented a methodology for identifying concrete cracks using F-CNN in a more challenging task with the presence of cracks and crack like noise patterns. Their binary classification of cracks and intact surfaces recorded an accuracy of 47%.

However, few studies have focused their scope into identifying the structural damage causes and degree of severity through DL image-based techniques. For instance, (Gao and Mosalam, 2018) presented a simplified version of ImageNet for structural engineering, named Structural ImageNet, with four baseline recognition tasks: component type identification, spalling condition check, damage level evaluation, and damage type determination. They have classified damage and predicted its degree based on cracks in the structural-level images. They designed two experiments based on two strategies to find the relative optimal model parameters. Their results showed that the testing accuracy was 77%

and 57.7% for the damage level and damage type, respectively, for the first classifier, and 89.7%, and 68.8% for the second one. This relatively low accuracy was attributed to overfitting problems. (Ye *et al.*, 2019) proposed a fully convolutional neural network (FCN) named Ci-Net for structural crack identification. Crack images from an indoor concrete beam test were adopted for validation of its structural crack recognition capacity. They reached an accuracy of 93.6%. (Xu *et al.*, 2019) proposed a modified faster region-based convolutional neural network (Faster R-CNN) for multitype seismic damage identification and localization (i.e., concrete cracking, concrete spalling, rebar exposure, and rebar buckling) of damaged reinforced concrete columns from Image. Rectangular bounding boxes were obtained to localize multitype structural damages along with the corresponding category labels and classification probabilities. Their test results showed that their trained network could automatically identify and localize multitype seismic damages with an average accuracy of 80%.

The various studies noted above-identified structural damage of reinforced concrete members based on the frequent use of DL based models, which consists of classifying the damage based on a dataset of images with the specified damage class. However, as reported in each study, there was always a margin of error and a lack of accuracy in classifying the type of structural damage acting upon the concrete surface and predicting its severity. Furthermore, labeling a dataset of images according to its damage form or pattern to predict the damage type and its level is a laborious and time-consuming task. This approach heavily relies on engineering judgment and is thus associated with high levels of subjectivity. In the current study, a novel technique to identify the kind of crack based on its orientation is proposed. This feature helps to determine the cause and severity of the damage.

Another knowledge gap in the existing literature is that there is a dearth of studies that explored rational crack quantification. For instance, (Beckman *et al.*, 2019) proposed an (F-RCNN) DL based automated volumetric damage quantification using a depth camera (3D scanner) only to quantify concrete spalling in terms of volume. Therefore, in the present study, precise quantification of crack features in terms of width, length, and angle of orientation is proposed based on mathematical and geometric operations to characterize

both structural and durability related damage of structural members and to assess its degree of severity in short computing time. Thus, the novelty of the present study consists of the following aspects: i) Developing a DL Image-based technique with high accuracy and less computing time; ii) Using the DL to propose a model to classify crack types in concrete based on their orientation in an automated way since previous studies used manually labeling techniques to identify the kind of damage, which is subjective, time-consuming, and laborious; iii) Quantifying concrete cracks in terms of length, width, and angle of orientation using a combination of DL and improved Otsu Image Processing Technique (IPTs); and iv) Identifying the severity of structural damage based on the allowed range of concrete crack widths for different structures, including buildings and bridges as per guidance from international standards and design codes.

This chapter is divided into three sections. The first illustrates the data manipulation. The second explains the input, architecture, and output of the DL classifier. The third deals with image processing and segmentation techniques used for the calculation of crack features, including length, width, and angle of orientation. The fourth section concerns the presentation of the results and their validation via available experimental results and on-site real concrete crack measurements.

3.1.1 Research Significance

Previous research generally reported DL classification of civil structures and infrastructures in qualitative terms, without calculating the important damage features of cracks in terms of width, length, and angle of orientation. In the present study, a modified Otsu image processing technique was combined with a DL classifier to localize, classify, and quantify cracks in cracked cement-based structural elements. Moreover, the nature of structural damage acting upon the structural member (flexural, shear, combined effect, corrosion of rebars) and its degree of severity was investigated using the allowed intervals of crack width limitations based on conventional worldwide building codes. The results should stimulate a critical look into the state-of-the-art of Image-based DL structural damage quantification and prediction protocols and highlight the need for a thorough analysis that can assist engineers in conducting rational concrete structures inspection.

3.2 Data Manipulation

3.2.1 Data Preparation

Images were retrieved from an open dataset available at (Özgenel, 2019), an annotated dataset available for training and testing of artificial intelligence-based crack detection and localization algorithms for concrete. The dataset is divided into cracked and safe images for image classification. It includes 20,000 cracked images and another 20,000 for safe concrete structures, at a resolution of 227x227 pixels with RGB channels. The dataset was generated from 458 high-resolution images (4032x3024 pixels) using the method proposed by (Zhang *et al.*, 2016). The dataset is pixel-based, which means that the full concentration in this study is based on close images taken at a short distance from the object (between 25 and 50 cm), presenting one single crack per Image. Other photos from various datasets in a pixel-level scale were provided from multiple datasets, such as SDNET2018 (Maguire *et al.*, 2018), Structural ImageNet (Gao *et al.*, 2018) and google and Baidu Images.

3.2.2 Data Processing

The first and foremost step to guarantee a robust classifier consists of data cleaning and visualization. Providing high-quality images with a variety of obstructions, including shadows, surface roughness, scaling, edges, holes, and background debris in the training set, is a crucial step to produce an accurate and realistic model. Low-quality images can dramatically affect the accuracy of the CNN classifier (Kannoja *et al.*, 2018). To address this issue, manual selection excluding any distorted, blurry, or low-resolution images has been set. Moreover, no edge cracks' pictures were selected since the Image will get smaller as much as it passes through the network, which implies that cracks on edges have less chance during training to be recognized by a network than those with cracks in the middle of images. Also, it is not possible to identify whether such crack features are cracks or not, which can, therefore, lead to the training data setting false annotations. All images were resized to an input image of 227×227 pixel resolution. This is because when the network is trained on relatively small images, it can scan any desired feature larger than the designed size, but not vice-versa.

3.2.3 Data Labelling

After being exposed to the CNN classifier, every Image loses its RGB channels and will be transformed to the grayscale level to reduce computational time since the color feature is not essential. The CNN classifier was trained using 10000 images, 5000 for cracked images, and 5000 for non-cracked images. A split of 60:20:20 for training, validation, and testing was applied. Hence, from those 10000 images, 2000 are for validation, 2000 for testing, and the rest is for training. Three sets of classification algorithms were applied. The first is a binary classifier that aims to detect whether the Image presents a safe structure (S) or a cracked one (C). The second is to differentiate between four classes of cracked concrete surfaces displaying 1200 images each: Vertical Left (VL), Vertical Right (VR), Horizontal Right (HR), and Horizontal Left (HL). The third classifier encompasses the classes together in a single classifier, presenting five classes in total: i) safe structure, ii) VL, iii) VR, iv) HR, and finally v) HL cracks, with 1200 images for each class. **Figure 3.1** explains how the labeling of the images was done. Every Image was composed of four parts. Based on the point of initiation of the crack in the picture, the four labels VR, VL, HL, and HR, were attributed.

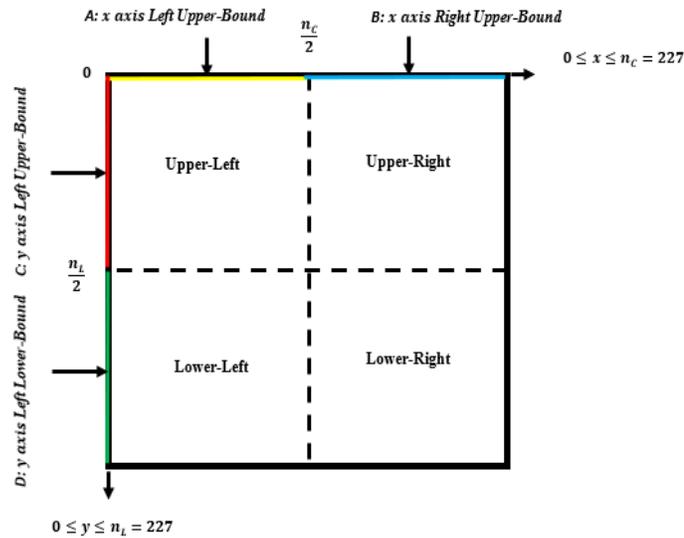


Figure 3.1: Method of Labelling Images According to Their Orientation.

Considering S_p as the starting point of the crack, which is the point at which the crack initiates, the attribution of these labels was done as follows:

$A = x \text{ axis Left Upper - Bound} = \{(x, y) | 0 \leq x \leq \frac{n_c}{2} \ \& \ y = 0\}$, If $S_p \in A$, the crack is considered as VL.

$B = x \text{ axis Right Upper - Bound} = \{(x, y) | \frac{n_c}{2} \leq x \leq n_c \ \& \ y = 0\}$, If $S_p \in B$, the crack is considered as VR.

$C = y \text{ axis Left Upper - Bound} = \{(x, y) | 0 \leq y \leq \frac{n_L}{2} \ \& \ x = 0\}$, If $S_p \in C$, the crack is considered as HL.

$D = y \text{ axis Left Lower - Bound} = \{(x, y) | \frac{n_L}{2} \leq y \leq n_L \ \& \ x = 0\}$, If $S_p \in D$, the crack is considered as HR.

Figure 3.2 shows an example of the attribution of these four labels. The third classifier was used for the rest of the process. After predicting the class and determining whether the Image presents cracks or not, a conditional block was introduced. If the picture showed a safe structure, the algorithm returns a message informing the user that the structure is secure, and there is no need for further calculation operations. However, if the Image presents cracks, the tested Image will be transferred to a MATLAB script to localize the crack in the Image using image segmentation techniques and to determine its geometrical properties, including its length, width, and angle of orientation.

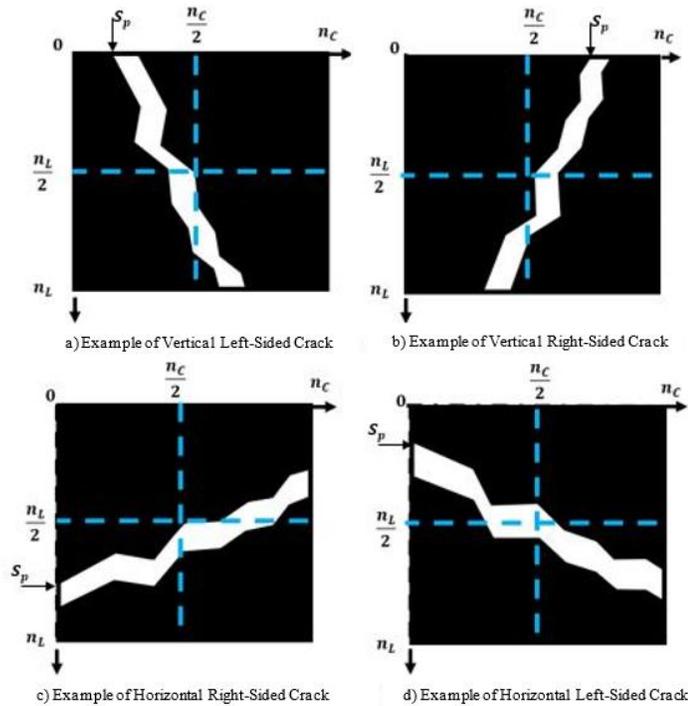


Figure 3.2: Example of Labeled Images According to the Proposed Method.

3.3 CNN Classification

The Convolutional Neural Network (CNN) architecture was designed based on a group of layers, including an input layer, convolutional layer, pooling layer, activation layer, and finally, the output layers. Other auxiliary layers like the dropout and batch normalization (BN) were introduced. **Figure 3.3** illustrates the configuration of the classifier. The first layer (Input layer) receives a $227 \times 227 \times 1$ image (a grayscale square image having 227 rows and columns). Then the input data pass through the architecture and are reduced to $1 \times 1 \times 64$. The vector, including the 64 elements, traverses the (ReLU) layer. In the end, the Softmax layer predicts the final output based on the kind of classification. The first classification is binary, where prediction consists of determining whether the Image contains cracks or not. However, the second is multi-purpose, where the classification aims to predict the orientation of the crack (VR, VL, HR, or HL), while the final classifier encompasses all these together. **Table 3.1** shows the detailed dimensions of each layer and operation.

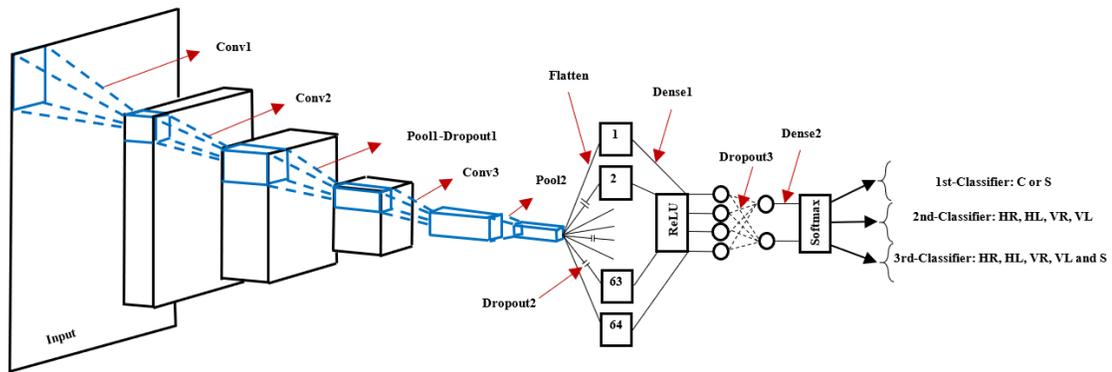


Figure 3.3: Overall Architecture of the CNN Classifier.

3.3.1 Convolution layer

Figure 3.4 shows a simple convolution operation introduced by (Goodfellow *et al.*, 2016). The convolutional layer carries the central portion of the network. A convolution is a linear operation that deals with the multiplication of a set of weights with the input. The multiplication is done between an array of input data and a two-dimensional array of weights called filter or kernel. The size of the filter is smaller than the input data; its

multiplication with the input data is a dot product. Dot product or scalar-multiplication is an element-wise multiplication between two 2D-arrays, which is then summed, resulting in a single value. During the forward pass, the filter moves along the width and height of the Image, producing the image representation of the receptive region. This two-dimensional representation is known as an activation map that shows the response of the filter at each spatial position of the Image.

Table 3.1: Dimensions of Layers and Operations.

| Layer | Height | Width | Depth | Num of Parameters |
|-----------------------------|------------|-------|--------|-------------------|
| Input | 227 | 227 | 1 | - |
| Conv1 | 227 | 227 | 32 | 320 |
| ReLU | 227 | 227 | 32 | 0 |
| Conv2 | 225 | 225 | 32 | 9248 |
| ReLU | 225 | 225 | 32 | 0 |
| Pool1 | 112 | 112 | 32 | 0 |
| Dropout1 | 112 | 112 | 32 | 0 |
| Conv3 | 110 | 110 | 64 | 18496 |
| ReLU | 110 | 110 | 64 | 0 |
| Pool2 | 55 | 55 | 64 | 0 |
| Dropout2 | 55 | 55 | 64 | 0 |
| Flatten | 1 | 1 | 193600 | 0 |
| Dense1 | 1 | 1 | 64 | 12390464 |
| ReLU | 1 | 1 | 64 | 0 |
| Dropout3 | 1 | 1 | 64 | 0 |
| Dense2 | 1 | 1 | 2 | 130 |
| Softmax | 1 | 1 | 2 | 0 |
| Total Parameters | 12,418,658 | | | |
| Trainable Parameters | 12,418,658 | | | |

The filter slides across the input image with a step called a stride. If L is the actual convolution layer, f^L is the size of the filter, p^L it is padding, s^L its stride and c^L is the number of filters. For an image having the size of $(n^{L-1}, m^{L-1}, c^{L-1})$, where n is the height of the Image, m the width and c the number of channels, the output of the layer would be a two-dimensional representation of the Image having the following size (n^L, m^L, c^L) , where n^L is computed by **Equation 3.1**, and m^L is calculated in **Equation 3.2**.

$$n^L = \frac{n^{L-1} + 2 \times p^L - f^L}{s^L} + 1 \quad (3.1)$$

$$m^L = \frac{m^{L-1} + 2 \times p^L - f^L}{s^L} + 1 \quad (3.2)$$

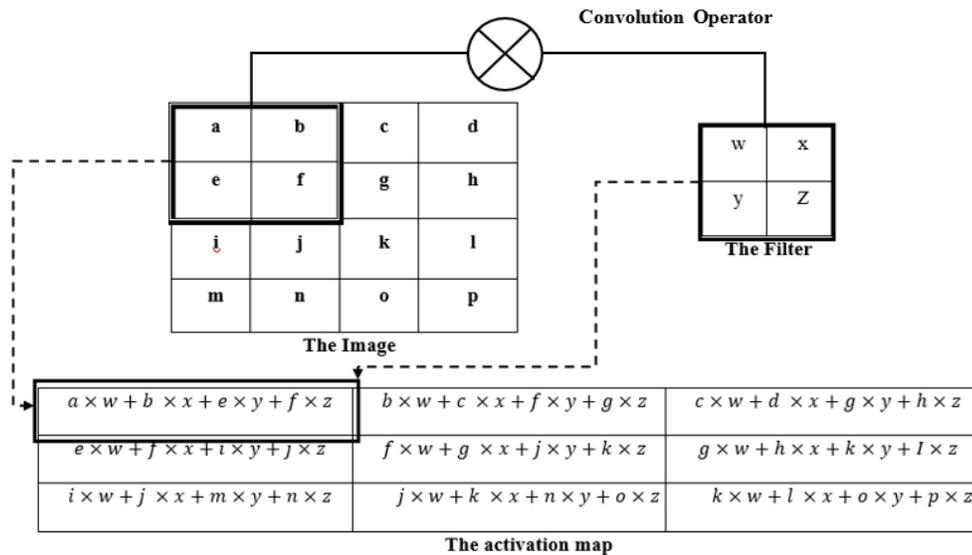


Figure 3.4: Convolution Operation.

3.3.2 Pooling layer

Figure 3.5 illustrates a simplified, pooling example. The pooling layer helps to reduce the spatial size of the representation, which decreases the number of iterations, computation size, and weights. Several options for pooling could be presented, such as the average of the rectangular neighborhood, L2 norm of the rectangular neighborhood, and most commonly max pooling, which gets the max values from the subarrays of an input array. In a pooling operation, for an activation map having size $n \times m \times c$, a pooling filter of a

size f and stride s would lead to a reduction in the spatial size of the output layer, following the two-formula listed below:

$$n^{output} = \frac{n - f}{s} + 1 \quad (3.3)$$

$$m^{output} = \frac{m - f}{s} + 1 \quad (3.4)$$

Where the output consists of a volume of size $(n^{output}, m^{output}, c)$ with n^{output} is computed by **Equation 3.3**, and m^{output} is calculated in **Equation 3.4**.

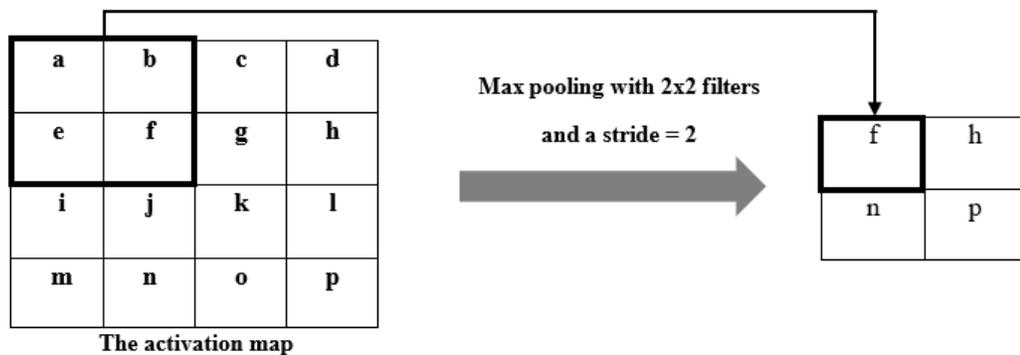


Figure 3.5: Pooling Operation.

3.3.3 Non-Linearity or Activation layers

A non-linear ReLU function was used as an activation function, as shown in **Equation 3.5** below:

$$f(x) = \max(0, x) \quad (3.5)$$

3.3.4 Dropout layers

Dropout is one of the techniques used to tackle the problem of overfitting for neural networks and to generate much more efficient training examples by reducing the coadaptation between neurons. Overfitting is determined whenever the training loss is way smaller than the testing loss. The core idea behind dropout is to disconnect the connections

between neurons randomly, having a fixed dropout rate, as presented in **Figure 3.6**. A dropout coefficient of 0.5 was used in the present study.

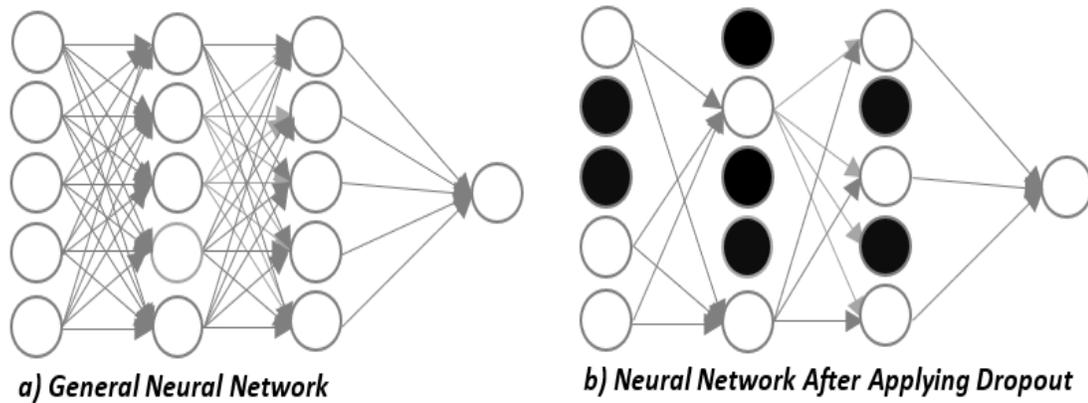


Figure 3.6: Dropout Operation.

3.3.5 Batch Normalization

Figure 3.7 shows an example of the application of Batch Normalization (BN) to an activation x over a mini batch. BN is a technique used to improve the speed and performance of a neural network as well as to produce more reliable models. (Ioffe *et al.*, 2015) proposed BN to mitigate the problem of the internal covariate shift. During training, as moving forward, the parameters of the preceding layers changes, the distribution of inputs changes accordingly in a way that the current layer needs to be always updated and readjusted to the actual distributions. BN normalizes the output of the previous activation layer performing the subtraction of the batch mean and the division by the batch standard deviation. Following this shift, the weights in the next layers are no longer optimal. Thus, BN adds two parameters to each layer so that the normalized output will be multiplied by a parameter for the standard deviation and adds a mean parameter.

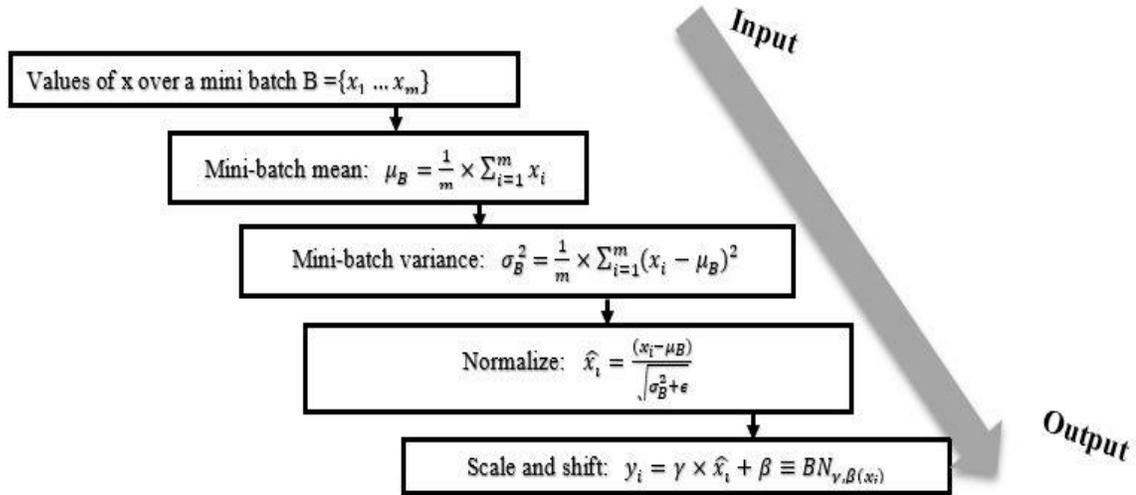


Figure 3.7: Batch Normalization Transform Applied to an Activation x Over a Mini Batch

Following this method, BN allows Stochastic Gradient Descent (SGD) to deformalize the previous layer by modifying these two weights for each activation, rather than performing it to the entire weights of the network, which can disturb the network and lose its stability.

3.3.6 Softmax Layer

The Softmax layer is used to find the most probable occurrence of a classification, where the probability of a class is maximum. The softmax layer is used just before the output layer of CNN. It outputs a probability distribution where the sum of the output values is equal to one. The Softmax function is given by $P(y^{(i)} = n | x^{(i)}; W)$ and is computed by **Equation 3.6**, where I varies from 1 to m : number of training examples, j is the class out of n classes, W is the adopted weights, and finally $W_n^T x^{(i)}$ serves as the layer input. The function returns the probability per each class of the input.

$$P(y^{(i)} = n | x^{(i)}; W) = \begin{bmatrix} P(y^{(i)} = 1 | x^{(i)}; W) \\ P(y^{(i)} = 2 | x^{(i)}; W) \\ \vdots \\ P(y^{(i)} = n | x^{(i)}; W) \end{bmatrix} = \frac{1}{\sum_{j=1}^n e^{W_j^T x^{(i)}}} \times \begin{bmatrix} e^{W_1^T x^{(i)}} \\ e^{W_2^T x^{(i)}} \\ \vdots \\ e^{W_n^T x^{(i)}} \end{bmatrix} \quad (3.6)$$

The adopted n (number of classes) in this study is equal to five ($n = 5$), presenting the safe, VR, VL, HR, and HL class. While the number of training examples is $m = 1200$, those parameters were selected using a trial and error process. The start used 5000 images for each of the five classes. However, it was observed that the accuracy of classification was still high (above 95%), even for 1200 images. Thus, to reduce the computation time, m was adopted to be 1200.

3.3.7 CNN Classifier Considerations

The same CNN architecture and hyperparameters (learning rate, batch-size, optimization method, loss, number of layers, size of the stride for the pooling operation, size of the filter for the convolution operation, etc.) for the three models were adopted. The only difference is in the number of data available for each class. For example, 5000 images per class were provided for the first classifier; however, for the second and third classifiers, data was less and needed more labeling effort depending on its orientation.

To seek better performance, data shuffling was applied. Data shuffling is a procedure that serves to reduce the variance of a statistical distribution and prevent overfitting by redistributing the data across its classes or targets. The adopted learning rate is $l_r = 0.0001$. It is defined as a hyperparameter that controls the extent of change of a specific model in response to the estimated error each time the weights of the model are updated. The modal performance is highly dependent on choosing the l_r . If its value is too small, a longer training process is estimated. Otherwise, the model may run too fast, affecting the CNNs performance and leading to an unstable training process.

The selected momentum was Nesterov, and its value is equal to $m = 0.9$. Nesterov momentum or Nesterov Accelerated Gradient (NAG) is a slight deformation of the standard gradient descent. It is used to determine in which direction the loss is low, which helps to speed up the training and significantly improves convergence.

The loss function is defined as a function that outputs the values of one or more variables of the network onto a real number that indicates how well those parameters can fulfill the task that the CNN is intended to do. For binary classification (1st classifier), binary cross-

entropy was used, while Categorical cross-entropy was used for the two other classifiers (2nd and 3rd classifiers).

The number of filters was set to 32 for the first convolutional block having a 3x3 size, the stride was 1, and padding was unchanged. On the other hand, the second convolutional block was made of 64 filters with a 3x3 size and the same stride. In a convolution operation, a stride denotes the number of steps in which the filter is moved along the activation map; its default value is 1. **Figure 3.4** presents a stride of 2 for the corresponding convolution operation. In other words, the padding is a technique to conserve the same dimension of output as input after the convolutional operation. It consists of adding zeros to the input matrix (activation map) in a symmetric way.

The pooling layer had a stride equal to 2, for both first and second blocks. The batch size was chosen to be 32, and 100 epochs were selected. The time used to train this network was two to three hours for a typical CPU processor and a couple of minutes on GPU. All the described tasks in this study are performed on Google Colab (Carneiro *et al.*, 2018).

3.4 Image Segmentation

Image segmentation is the process of dividing a digital image into multiple segments or pixels. Its main goal is to simplify the representation of the Image and make it easier by locating objects and boundaries such as lines and curves in images. The output of image segmentation is a group of segments that consistently cover the entire Image. Various algorithms have been applied for image segmentation (Yuheng *et al.*, 2017). These include methods of thresholding, clustering, motion, and interactive segmentation, compression-based methods, histogram-based methods, edge detection, dual clustering, region growing, partial equation-based methods, variational methods, graph partitioning methods, watershed transformation model-based segmentation, multi-scale segmentation, semi-automatic segmentation, and trainable segmentation.

After testing a new image using the third CNN classifier, one of the five categories of Safe, VR, VL, HR, or HL would be generated. A conditional block is then located at the end of the CNN classifier, telling the user that the structure is safe and does not need rehabilitation

for the current time. However, if one of the other classes is generated, all subsequent parts (**Sections 3.4, 3.5 and, 3.6**) will be considered. Edge detection and thresholding are the segmentation techniques used herein to process the tested Image.

3.4.1 Image Improvement

Before applying thresholding, the tested Image needs to be improved, for instance, to remove any background noise or illumination effects and to suppress unwanted information from the image data, which makes essential Image features more detectable by the proposed algorithm. Several factors could lead to image disturbance, including non-uniform lighting, weather conditions, and low contrast between distress and background. To perform image improvement, a non-linear filter considering the mean and variance of local grey values was used to remove the non-uniform background intensity. Reducing the noise in an image can be challenging since useful details in an image could be removed. Thus, the choice of the filter must be precise. The non-linear filter is computed by **Equation 3.7**.

$$f^* = Z(i, j) \times [f_{org}(i, j) - f_{blur}(i, j)] + m1 \quad (3.7)$$

Where, f^* is the frequency of the filtered Image, $Z(i, j)$ is the local gain factor, which equals to 1, $f_{org}(i, j)$ is the original frequency of the Image, $f_{blur}(i, j)$ is the frequency of the blurred Image, and $m1$ is the mean value of the original Image. The blurred Image presented in **Equation 3.7** and computed by **Equation 3.8** is obtained by convoluting a gaussian 9x9 low pass spatial filter with the original image, as shown below:

$$f_{blur}(i, j) = f_{org}(i, j) \times F^{-1} \left(H(\Omega_i, \Omega_j) \right) \quad (3.8)$$

$$H(\Omega_i, \Omega_j) = \exp \left(- \frac{D^2(\Omega_i, \Omega_j)}{2 \times \sigma_0^2} \right) \quad (3.9)$$

where $H(\Omega_i, \Omega_j)$ is the Gaussian Transfer Function computed in **Equation 3.9** and σ_0 is the cut-off frequency.

3.4.2 Thresholding using Otsu's Method

Thresholding is a technique used for image segmentation. It consists of creating a histogram of the gray level values to be used for determining the peaks that exist in an image. Concrete cracks generally include an abrupt change in the gray level of two adjacent regions of variant gray levels. An adequate threshold is extracted based on the means of these two regions. In this study, Otsu's method proposed by (Sun *et al.* 2009) and improved by (Hoang, 2018) was computed by **Equation 3.10** to perform image segmentation. It is a nonparametric and unsupervised method of automatic threshold selection for picture segmentation. The optimal threshold was selected by the discriminant criterion, which helps to maximize the separability of the resultant classes in gray levels. The property of self-similarity is considered in this study, which can be translated by the presence of a single crack per Image. Otsu's concept is straight-forward. It consists of returning a single intensity threshold that separates pixels into two classes, foreground, and background. This method looks for the threshold that lowers the intra-class variance, defined as the weighted sum of variances for the two classes.

$$\sigma_w^2 = w_0(t) \times \sigma_0^2(t) + w_1(t) \times \sigma_1^2(t) \quad (3.10)$$

Where w_0 is computed by **Equation 3.11**, w_1 by **Equation 3.12**, σ_0^2 and σ_1^2 are the probabilities and variances of the two classes, respectively, and t is the threshold that separates between the two probabilities. The class probability $w_{0,1}(t)$ is computed from the Lb bins (a histogram is made up of bins, each bin represents a specific intensity value range) as follows:

$$w_0(t) = \sum_{i=0}^{t-1} p(i) \quad (11)$$

$$w_1(t) = \sum_{i=t}^{L-1} p(i) \quad (12)$$

MATLAB offers a built-in function called "graythresh" that returns the global threshold from a grayscale image, which is used in this study. After enhancement, the grayscale

Image would be transformed into a binary one using the MATLAB built-in function "im2bw".

3.4.3 Noise Removal and Image Connection

Even after enhancement, the tested Image may still be prone to some disturbance. The various noise removal steps implemented in this study are outlined below. For instance, some gaps between cracks could persist after image enhancement. To fill such gaps, a technique called closing operation computed by **Equation 3.13**, which is a standard mathematical morphology operator was used, employing some morphological transformations such as dilation and erosion. Mathematical morphology is often used for processing geometrical structures based on their typology and random functions. Many morphological transforms are built from basic morphological operations such as dilation and erosion:

$$A \cdot B = (A \oplus B) \ominus B \quad (3.13)$$

Where \oplus is the dilation operator and \ominus is erosion operator.

After dilating the Image, some image parts would become unconnected; *bwmorph* a MATLAB command was used in this chapter to connect disconnected close parts in images, performing the so-called "bridge operation," which aims to connect unconnected pixels whose values are set to 0 and change them to 1, if they have two non-zero neighbors that are not connected, as in **Equation 3.14**.

$$\begin{array}{cccccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & \rightarrow & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{array} \quad (3.14)$$

This operation was conducted until there was no change in the Image using $n = inf$, where n corresponds to the number of times the operation was applied. Despite connecting between the close parts in an image, some small spaces or "holes" could persist. To fill these holes, *imfill* was introduced in the binary image. After dealing with those spaces, only the largest object will remain, removing all small objects (isolated pixels) based on a specific value of pixels. After preprocessing, cracks would present some breakpoints which

could influence the crack localization task. The main reason for the creation of breakpoints is the discontinuity of the previous step caused by the changing of crack pixels into the background after thresholding or being removed after noise reduction. To connect between breakpoints, *bwconcomp*, a built-in MATLAB function was used.

3.4.4 Calculating Crack Dimensions

Before starting to calculate the dimensions of the crack (length, width, and angle of orientation), the tested Image is processed following the steps described in Section 3.4. **Figure 3.8-a** shows the original Image containing the crack, while **Figure 3.8-b** shows the Image converted to a binary scale passing through all steps described earlier.

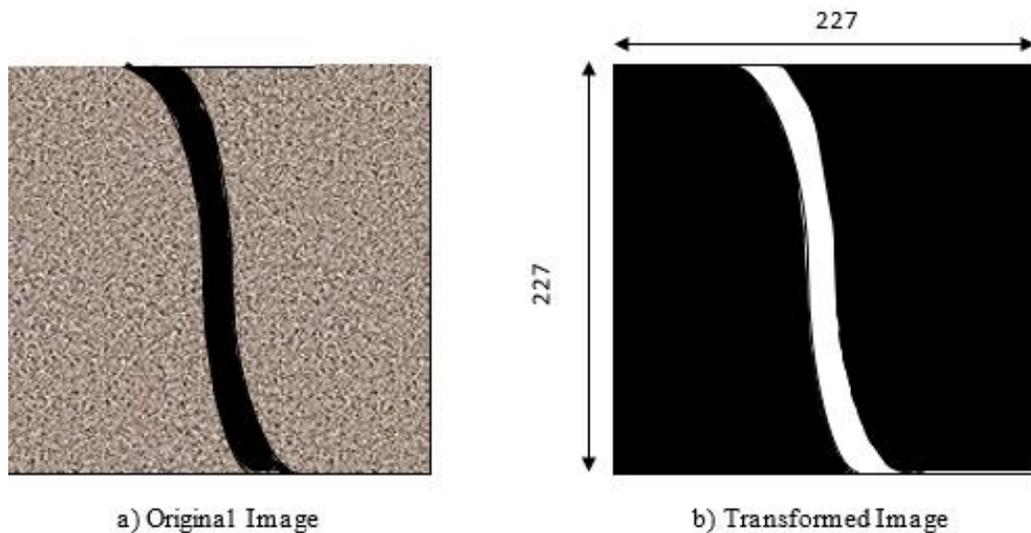


Figure 3.8: An Example of Image Transformation.

3.4.4.1 Length of Crack

To calculate the length of a crack, the first crucial step consists of determining the exact localization of the crack. This can be done by applying a built-in function in MATLAB called *bwboundaries*. Accordingly, the boundaries of the crack, as indicated with the arrows (*A*, *B*, *C*, and *D*) in **Figure 3.9**, would be considered so that the crack length is represented as the maximum distance between those boundaries. Considering $d_{i,j}$ as the distance between two points i and j , $d_{A,D}$ is the longest distance between every boundary point within the position of the crack.

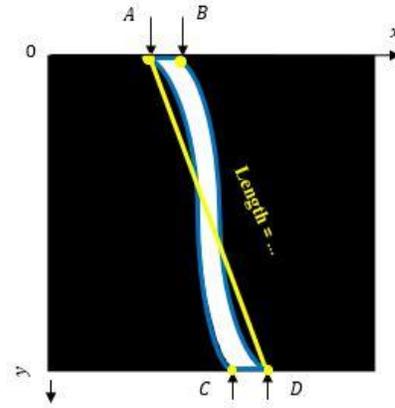


Figure 3.9: Length Calculation of the Crack.

Depending on the orientation of the crack in the Image, a comparison is made between each of the following distances to compute the maximum length of the crack. After comparing each of the following distances, $d_{A,B}$, $d_{A,C}$, $d_{A,D}$, $d_{B,C}$ and $d_{B,D}$ The length of the crack is supposed to be the biggest among all of those distances. All such values are calculated using the formula for Euclidean distance $pdist2$, computed in **Equation 3.15**, where the distance between two points A and B are equal to:

$$d_{A,B} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \quad (3.15)$$

Where x_A and x_B are located between 0 and the number of columns n_c (width of the resized Image), while y_A and y_B are between 0 and the number of rows n_r (height of the Image). In this study, $n_c = n_r = 227$ pixels.

3.4.4.2 Width of Crack

The width of a crack plays a vital role in predicting the degree of the structural damage it induces in a structure. The crack width is supposed to be the maximum distance between two points located on the crack boundaries. **Figure 3.10** indicates the process of calculating the width of the crack. This is based on the orientation of the crack, whether vertical or horizontal, and the boundaries that are created in the previous step to calculate the length of the crack. A loop is implemented to pass through the height of the Image (if the crack is vertical) and its width (if the crack is horizontal). After calculating those distances, an array

is created to contain them. The maximum distance corresponds to the maximum width, which is the corresponding width of the crack.

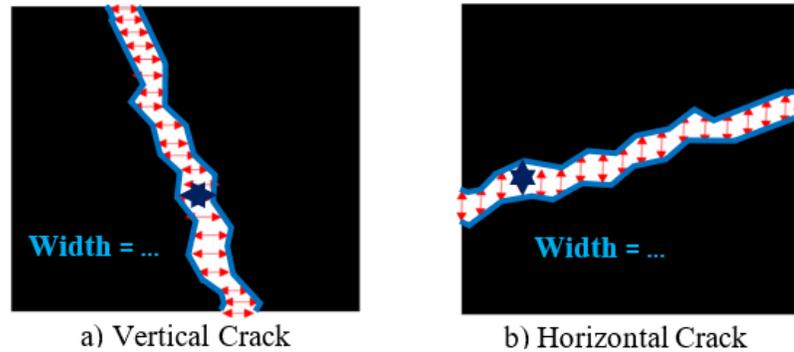


Figure 3.10: Width Calculation of the Crack.

3.4.4.3 The angle of Crack Orientation

The angle of orientation of the crack is defined as the angle located between the mainline passing through the crack that defines its length and the horizontal dashed line that starts from the starting point defined as S_p . The position of S_p is determined to depend on the location of the crack in the Image, whether horizontal, vertical, right, or left-sided, as illustrated in **Figure 3.11**. An example calculation of the value of the crack angle is defined in degree and illustrated in **Figure 3.12**. The angle of orientation of the crack is computed by the formula presented in **Equation 3.16**.

$$\alpha = \cos^{-1} \left(\frac{y_D - y_A}{d_{A,D}} \right) \quad (3.16)$$

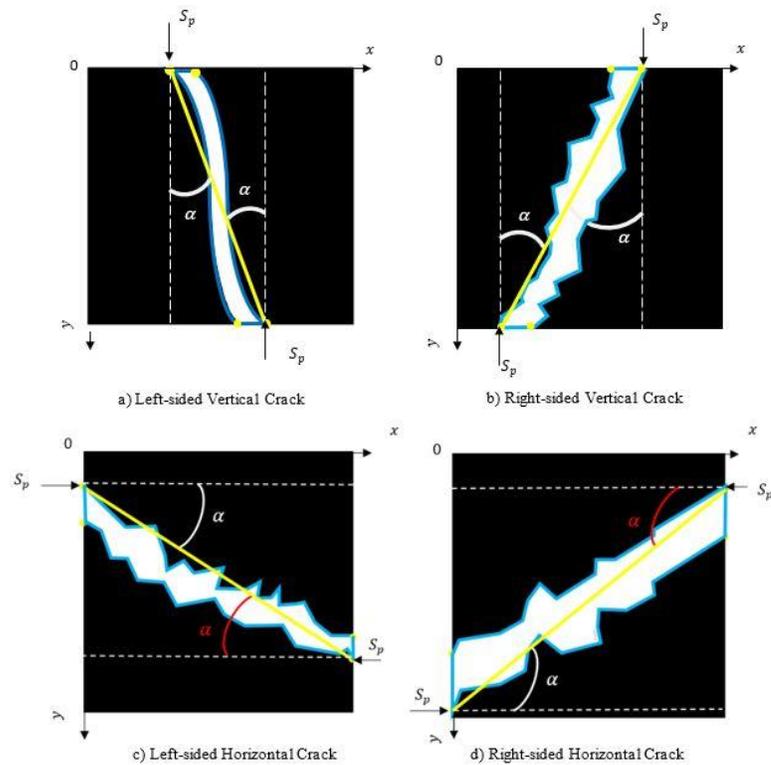


Figure 3.11: Configuration of the Angle of Orientation in the Image.

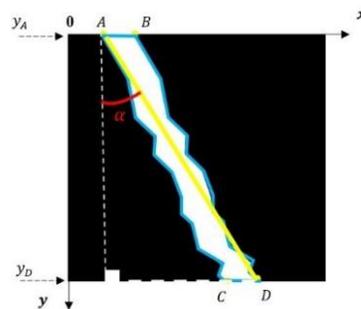


Figure 3.12: Calculation of Angle of Orientation of Crack.

3.4.5 IPTs Considerations

The crack length (**Section 3.4.4.1**) and the width (**Section 3.4.4.2**) are expressed in pixel. To transform these into units of meter, the user needs to introduce the area of the selected surface when shooting the picture ($A = a \times b$ in m^2). Subsequently, the algorithm makes the transformation respecting the Image's aspect ratio, based on **Equation 3.17**.

$$d_{A,B} (m) = d_{A,B} (Pixels) \times \sqrt{\frac{A}{n \times m}} \quad (3.17)$$

Where n is the number of columns and m is the number of rows, expressed in pixels. **Figure 3.13** explains the procedure of capturing images and defining their surface in (m^2).

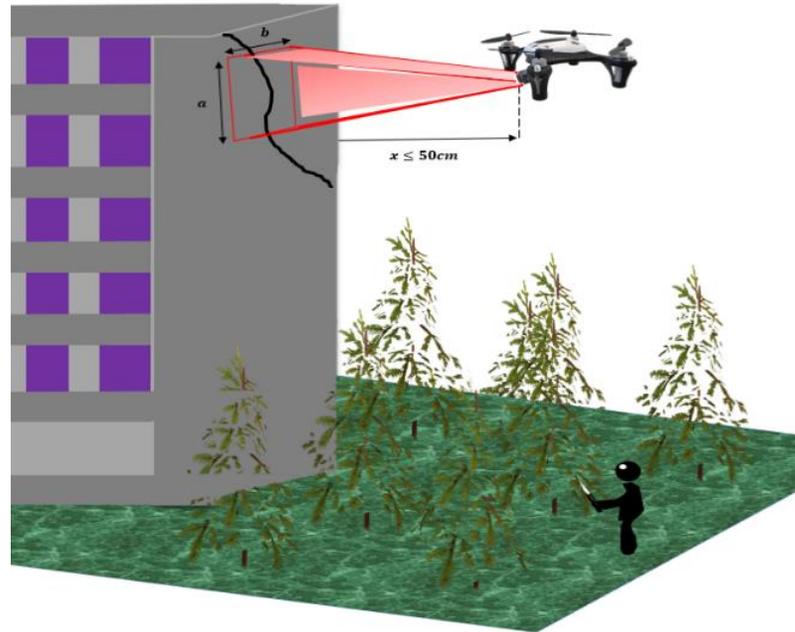


Figure 3.13: Introducing the Area of the Selected Surface.

3.5 Mode of Failure Prediction and Damage Severity Check

3.5.1 Prediction of Mode of Failure

The occurrence of cracking indicates that the material has been stressed beyond its strain capacity. For instance, reinforced concrete is stressed through the action of external loads, thermal and moisture gradients, chemical reactions, and enforced deformations. The topic of understanding how cracking takes place, its form, the significance of its timing, and its triggers is complicated since such various factors may intensify or counteract their mutual effects. In reinforced concrete structures, cracking appears as a solitary or a pattern phenomenon. Each crack can be characterized by some features, including its width, length, direction, and size. These features help to identify the cause and severity of the

damage. Cracking may originate during various phases of a building's life (design, construction, and service phases). Even though it is not possible to conclusively attribute the cause of cracking based on visual observation alone, specific characteristics may be associated with the nature of the underlying stress.

After predicting the orientation of a crack using the DL classifier, this information is processed using the Otsu IPTs to quantify the crack in terms of its length, width, and based on that, the angle of orientation of the crack is calculated. **Table 3.2** illustrates the prediction of the structural type of damage based on the actual value of the angle of orientation. **Figure 3.14** shows different types of cracks based on their orientation on a structural element.

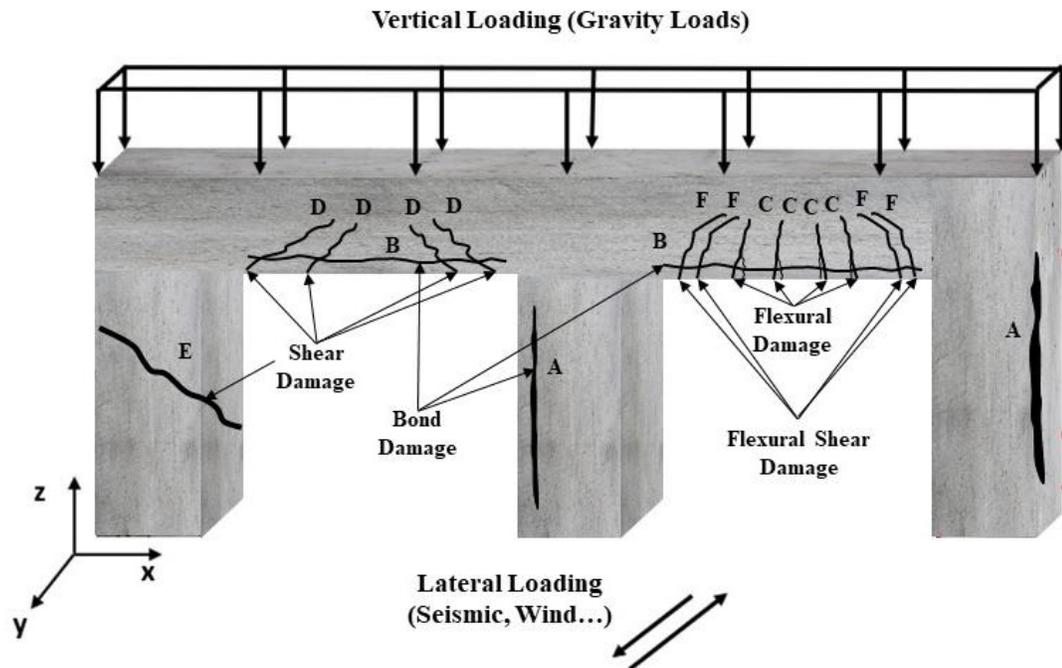
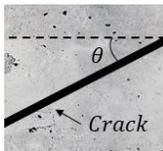
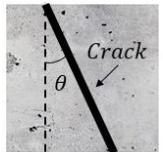


Figure 3.14: Causes of Crack Initiation in the Structural Elements of a Frame.

For instance, if the distribution of cracks is parallel to the reinforcement either in a column or a beam, or accompanied with spalling, it is likely a durability related damage, and it is considered as a bond cracking or corrosion of the reinforcement. It is then noted as a crack of type A for columns and type B for beams. However, flexural cracks usually originate at

the tensile face of the member, near its mid-span, and continue into, and often beyond, the reinforcement.

Table 3.2: Prediction of Mechanical Stress or Durability issues Causing Crack based on the Angle of Orientation.

| Structural element | Crack Angle | Crack orientation | Proximity | Crack creation causes |
|---|---|-------------------|-------------------------------------|-----------------------------------|
| Horizontal (Beam, Girder, Pier Cap) |  | [0, 5°] | Parallel to the longitudinal rebars | Corrosion |
| | | [30°, 50°] | Near the supports (ends) | Shear stress |
| | | [50°, 75°] | Between mid-span and supports | Combined Flexural Shear stress |
| | | [75°, 90°] | Close to mid-span | Flexural stress |
| Vertical (Wall, Pier, Column) |  | [0, 5°] | Parallel to the longitudinal rebars | Corrosion |
| | | [30°, 50°] | Near the supports (ends) | Shear stress |

These cracks would typically be at right angles to the tensile face of the member (crack Type C). A salient feature of shear cracking is that cracks are invariably inclined to the tensile face at an angle of approximately 45° (Richardson *et al.*, 2002). Shear cracks will often originate and stop within the boundary of the cracked face. A frequent location of shear cracking in beams is at the inner face of support, noted as crack type D. However, if most cracks occur in a diagonal direction for a vertical element such as a column, it is noted as Crack type E. But, if they form an "X" or "V" pattern, it is considered as shear-type damage, for instance mostly present in shear walls. In addition to the combined action of flexural-shear damage that may appear on the surface of a beam or any horizontal structural

member, between its mid-span and its supports, this type of cracking is noted as a crack of type F.

3.5.2 Damage Severity Check

Maximum values for design crack widths in prestressed and reinforced concrete members extracted from several design codes of practice are given in **Table 3.3**. Generally, allowable crack width varies in the interval of [0,1 mm] for the considered design codes. Any crack width that exceeds the specified values could compromise structural integrity. Although cracking in concrete structures is practically unavoidable, the limitation of its occurrence and extent can help to ensure structural safety and durability. **Figure 3.15** illustrates the required inputs for the algorithm, where the user is required to enter the area of the object being captured by the camera, the type of concrete structure (bridge, building, tunnel, etc.), and the structural component (column, beam, wall, etc.). Then the algorithm calculates the crack features, as explained in Sections 3 and 4. Subsequently, the algorithm indicates the mechanical cause of the crack (e.g., shear, flexural, corrosion, or combined action) and its degree of damage based on **Tables 3.2** and **3.3**, depending on the provided structural element and type of concrete structure.

For instance, if the structure is a building, and the surface of the structural element being processed is a beam (horizontal element), then information about the kind of structural damage (flexural, shear, combination, corrosion) and the degree of severity based on the provided building codes (Japanese Society of Civil Engineers (JSCE), American Concrete Institute (ACI), Canadian Standards Association (CSA), Eurocode 2, New Zealand (NZ), and Australian (AS) codes are delivered. To validate the proposed algorithm and assess its accuracy, several images were tested. Images were taken at Western University (Structural Lab, Reinforced Concrete Frame, Spencer Engineering Building, etc.) under various lighting conditions on a pixel-level scale.

Table 3.3: Allowable range of concrete crack width based on current design codes with regards to exposure conditions and loading category for reinforced and prestressed concrete structures.

| Structural Type | Region | Country | Conventional codes | Allowable Crack Width (mm) |
|------------------------|---------------|----------------|--|-----------------------------------|
| Buildings | North America | USA | ACI-224R (Table 4.1) | 0.1 - 0.41 |
| | | Canada | CSA A23.3 (A 8.1.4.2) | No specification |
| | Asia | Japan | JSCE (Table 8.3.2) | 0.1 – 0.4 |
| | | Australia | AS.3600 (Table 8.6.2.2) | 0.2 - 0.4 |
| | | New Zealand | NZS.3101 (Table C2.1) | 0 - 0.5 |
| | Europe | EU nations | EC2 (Table 7.1 N) | 0.2 - 0.4 |
| Bridges | North America | USA | AASHTO (A5.5, A5.6, and A5.7) | 0.22 – 0.43 |
| | | Canada | CSA S6:19 (Clause 8.12.3.1 Table 8.6) | 0.15 – 0.35 |
| | Asia | Japan | JSCE | 0.1 – 0.4 |
| | | Australia | AS.5100.5 (VicRoads Standard Specification Section 610 - Structural Concrete. Table 1) | 0.1 – 0.2 |
| | | New Zealand | NZS.3101.1&2 (Clause 2.4.4.2 and Table 4.1) | 0.1 – 0.35 |
| | Europe | EU nations | EC2 (EN-1992-1-1, 7.31 table 7.1N) | 0.2 – 0.4 |

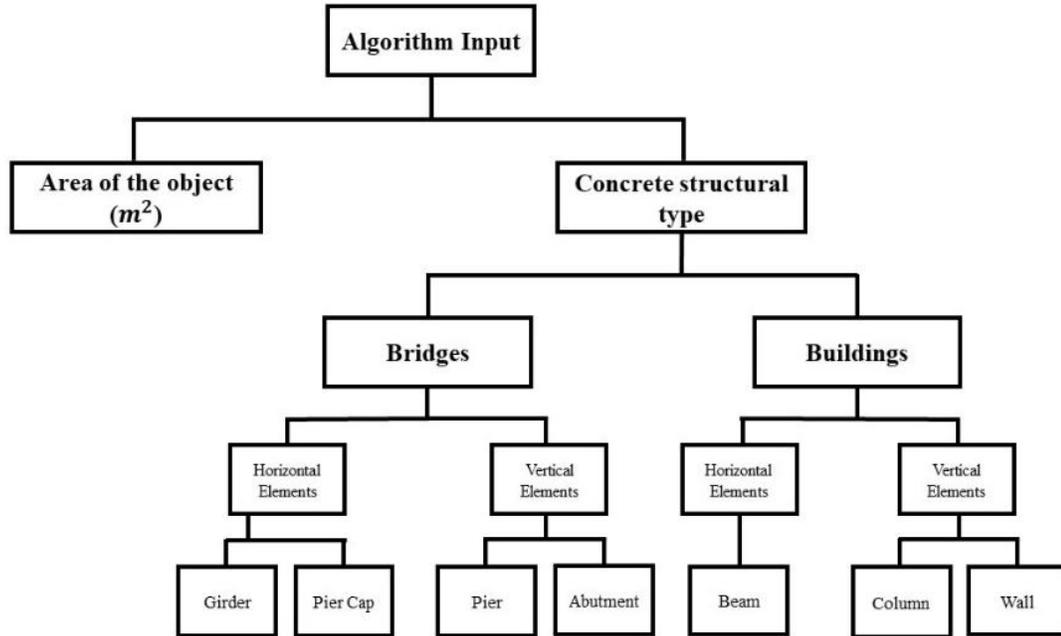


Figure 3.15: Database Inputs Model Incorporation.

3.6 Results and Discussion

The DL model was trained from scratch and performed very well on a multi-classification task based on the orientation of the crack, exceeding 97% accuracy. The IPT using the Otsu method was used to process the classified images by the third DL classifier to quantify the cracks in terms of width, length, and angle of orientation with a quantification error of 1.5%, 5%, and 2%, respectively. The proposed approach demonstrated its computation efficiency and its prompt performance promptly, unlike comparable methods reported in the current state-of-the-art, where damage recognition and quantification can be a time-consuming task, where the full process of obtaining crack width measurements needed almost 3.5 hours, and much longer for larger structures since the number of processed images would increase (Kim *et al.*, 2018).

The method proposed in the present study quantifies cracks in just three relatively simple steps (instead of five in the above approach): i) image acquisition; ii) classifying cracks according to their orientation in a pixel-based scale, and iii) quantifying cracks based on IPTs. The proposed algorithm performs the tasks in a short time of less than one minute.

Moreover, the Keras classifier for crack identification needs to be trained only once, then all the weight values, model architecture, and optimizer configuration will be saved in an HDF5 file (Folk *et al.*, 2011). The present study is thus expected to offer a nearly fully automated platform for accurate and timely monitoring of damage in reinforced concrete structures. It can be ready for implementation on unmanned aerial vehicles (UAV) for the vast engineering community.

Table 3.4: Best model performance for CNN classifications for 100 epochs calculation.

| Index of the classifier | Goal | Training | Validation | Testing | Training | Validation | Testing |
|-------------------------|--|--------------|--------------|--------------|----------|------------|---------|
| | | Accuracy (%) | Accuracy (%) | Accuracy (%) | Loss | Loss | Loss |
| 1 st | Safe or Cracked Concrete Surface | 99.57 | 98.6 | 98.25 | 0.015 | 0.051 | 0.057 |
| 2 nd | Orientation of crack VL, VR, HL And HR | 98.12 | 96.25 | 97.18 | 0.047 | 0.208 | 0.153 |
| 3 rd | First and Second Classifier combined | 97.63 | 96.5 | 96.17 | 0.0602 | 0.221 | 0.14 |

Figure 3.16 plots the training and validation accuracy of the three classification tasks for 100 epochs. The first observation is that the computation converges rather rapidly for the early 10 to 20 epochs. The proposed model demonstrated superior performance and adaptability, as indicated in **Table 3.4**. The recorded training accuracy was 99.57%, 98.12%, and 97.63 for training the first, second, and third classifiers, respectively. The testing results confirmed excellent performance. Testing accuracy of 98.25%, 97.18%, and 96.17% was recorded for the three classifiers, respectively. Another indicator of the network's performance is the low values of loss. To optimize the parameter values in a Neural Network (NN) architecture, a loss function is used. The loss function maps a set of parameter values for the network onto a scalar value that indicates how well those parameters accomplish the task which the network is intended to do.

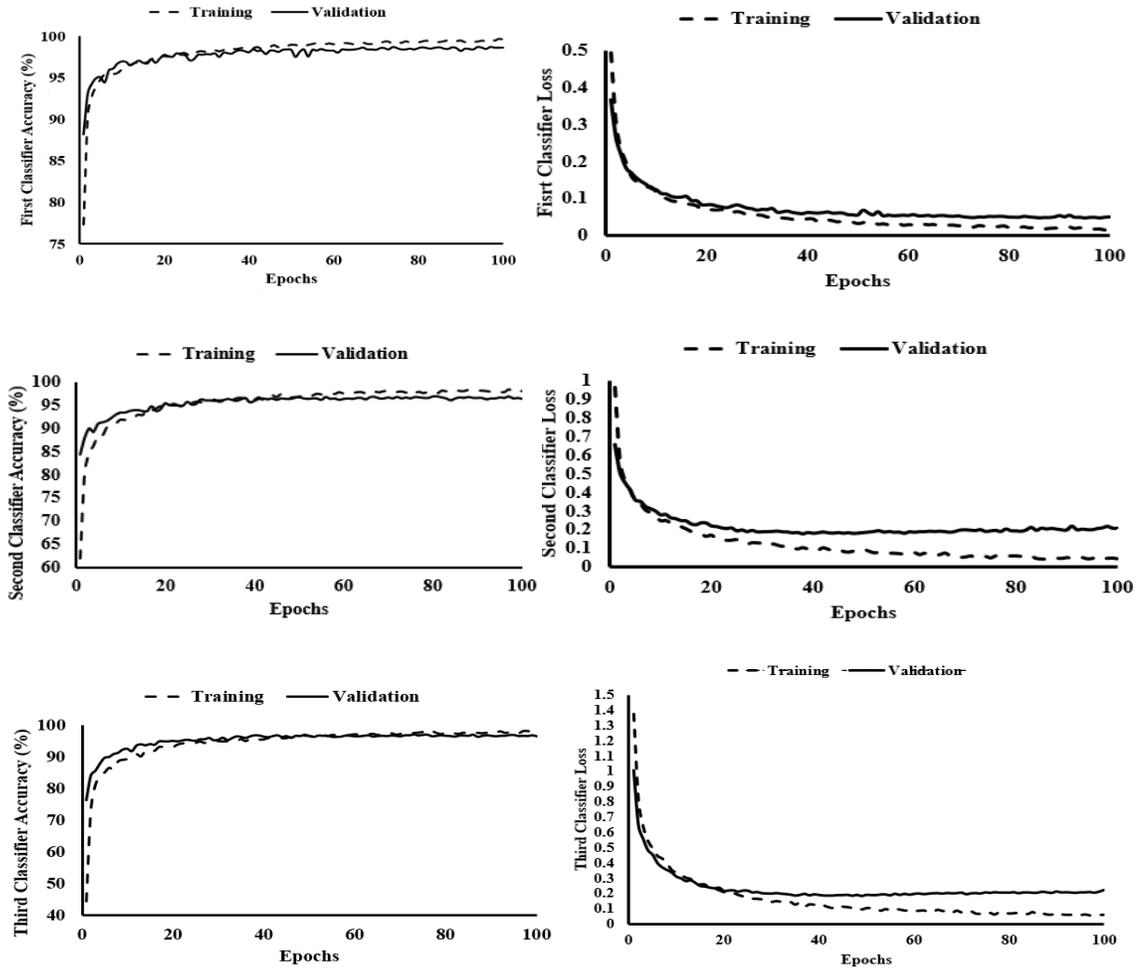


Figure 3.16: Accuracy and Loss Histories for the Three Recognition Tasks.

The obtained low values of loss using the categorical cross-entropy method demonstrate that overfitting problems did not influence this classifier. For instance, the obtained values of loss for the third classifier were only 6.02% and 14% for training and testing tests, respectively.

Moreover, the value of loss for training and testing increased whenever the classification task became harder. In other words, the obtained values of loss for the binary classifier and the second classifier, including the four classes, were lower than the loss values of the adopted classifier (including five classes as explained in Section 3). The loss of the first classifier was 1.5% for training, 5.1% for validation, and 5.7% for testing.

An approach called "Confusion Matrix" or Error Matrix was proposed to evaluate classification results since the precision of the classification for the prediction task is not reliable if the set of inputs contains more than two classes (Kohavi *et al.*, 1998). The confusion matrix is a table layout used to visualize the performance of an algorithm. Each line of this table represents the instances of a predicted class, while each column represents the actual class. Accordingly, three non-normalized confusion matrixes for all classification tasks with a probability of correct or incorrect predictions and values broken down by class are presented in **Figure 3.17**.

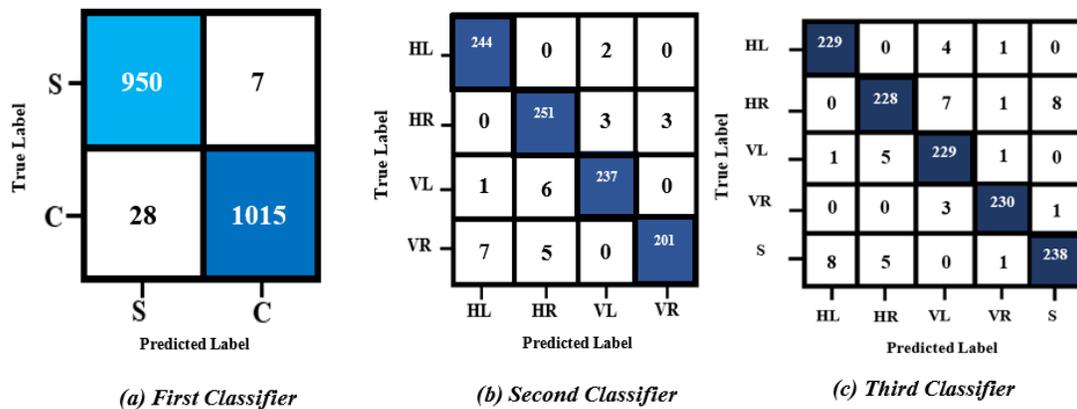


Figure 3.17: Unmanned Confusion Matrix of Test Prediction in the Three Tasks.

The confusion matrix was applied to the testing set, including 2000 images for the first classifier, 960 for the second, and 1200 for the third. All the classification tasks performed quite well with highly accurate predictions and low misclassification errors. The results indicate superior generalization performance of the Keras type architecture. **Figures 3.18, 3.19, 3.20, and 3.21** illustrate field test examples conducted at Western University, Canada. The examples were used to validate the proposed algorithm. For instance, **Figure 3.19** shows the real dimensions of crack captured on the beam due to flexural stress. A bounding box was drawn on the surface of the beam to indicate the part of the image that will be shot and preprocessed moving forward into the algorithm.

Table 3.5 compares the experimental and numerical values for the provided images, indicating nearly identical matching.

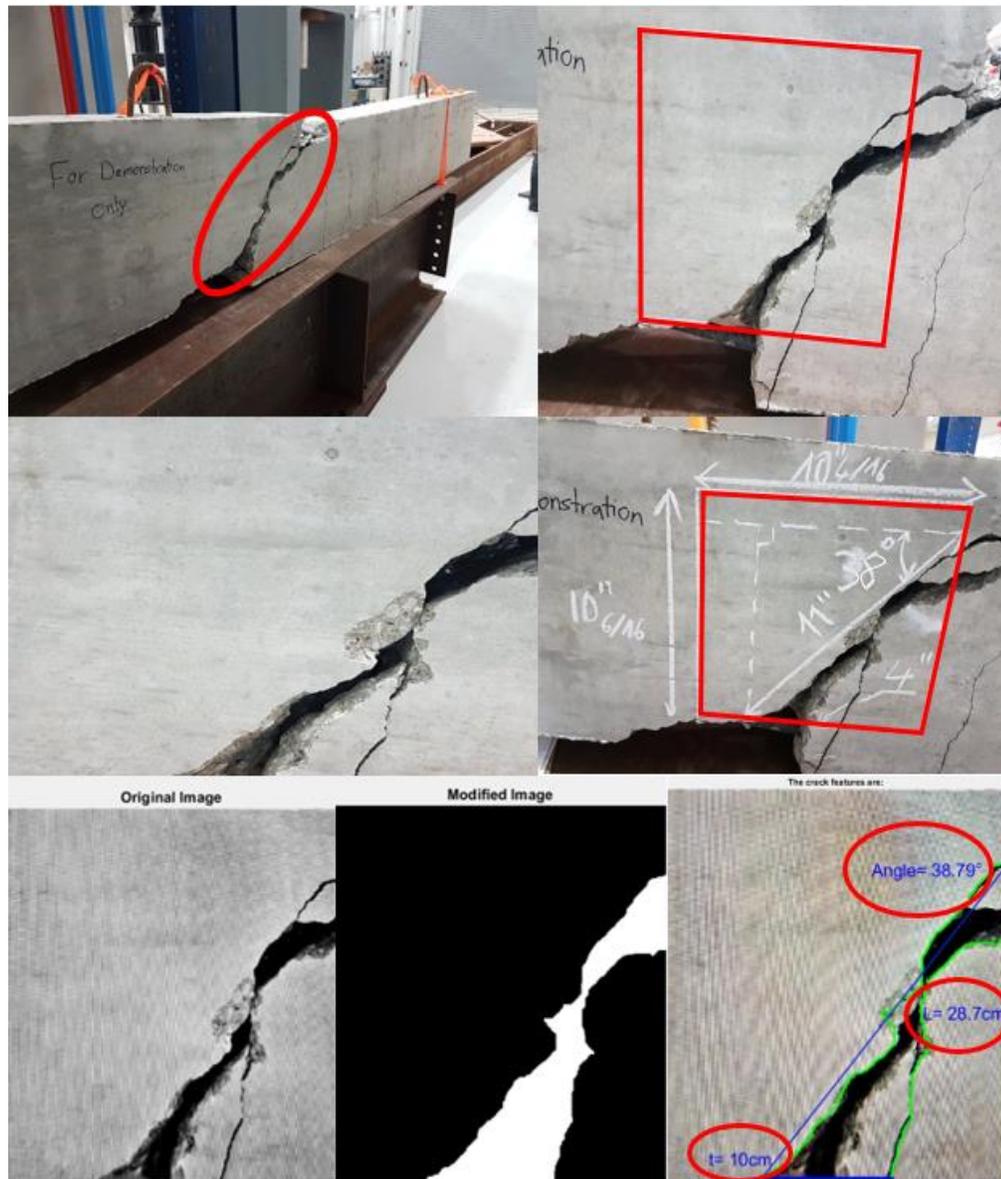


Figure 3.18: Calculation of Crack Features of the Damaged Beam under the effect of Shear Stress.

The error in percentage was computed for each example and then averaged for all the presented cases. The crack quantification algorithm, which was verified via a small-scale field test, provided a relative error of 1.5%, 5%, and 2% for the global calculation of the crack length, width, and angle of orientation, respectively.

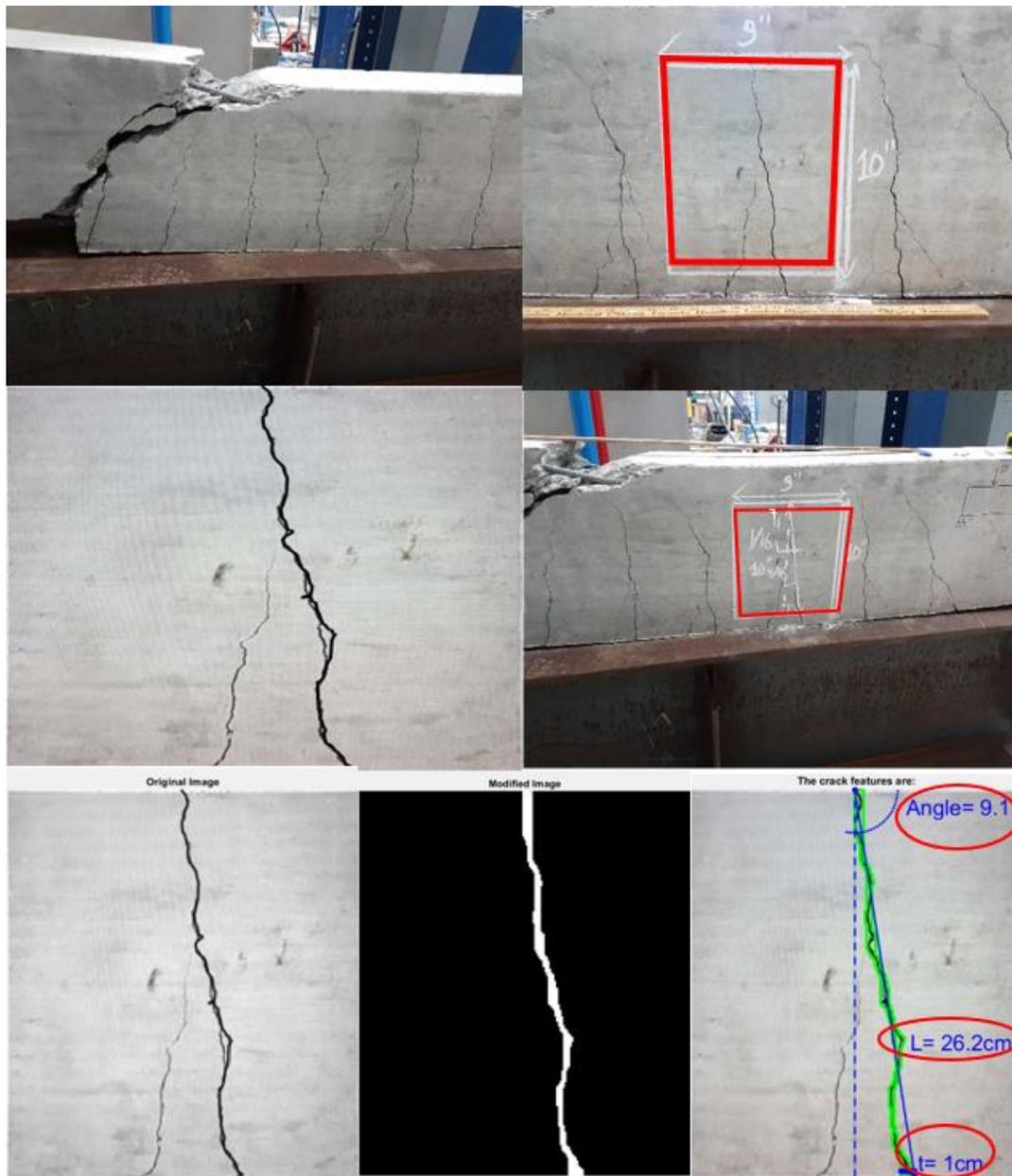


Figure 3.19: Calculation of Crack Features of the Damaged Beam under the Effect of Flexural Stress.

It is notable that the prediction of the mechanical stress causing damage and its degree for the different concrete structures considered herein perfectly matched the actual results, which further affirms the reliability of the proposed inspection algorithm.

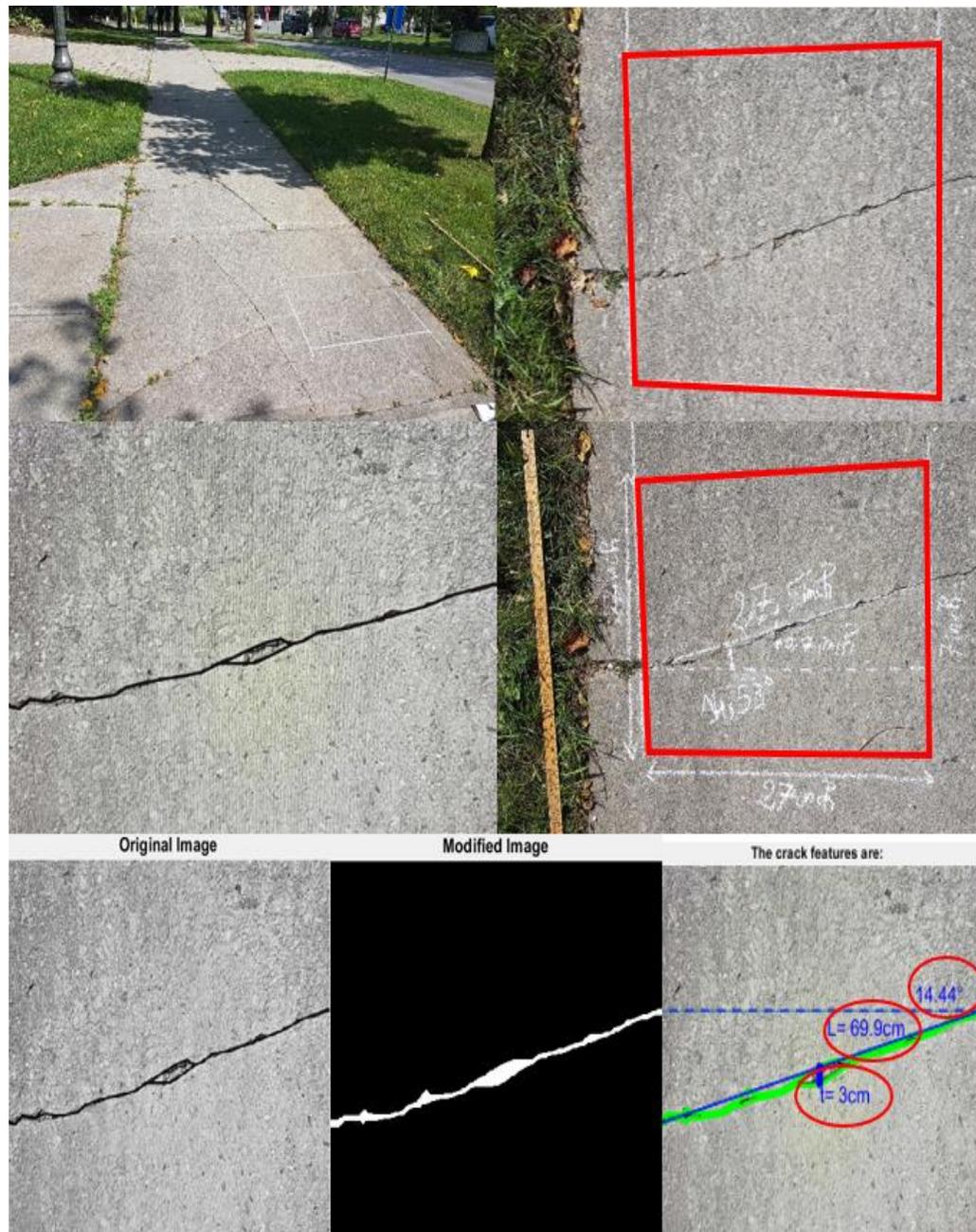


Figure 3.20: Calculation of Crack Features of Damaged Pavement under the Effect of uplifting of Tree Roots.

Several studies have affirmed that CNNs are a powerful tool in classification tasks. For instance, (Cha *et al.*, 2017) trained a CNN using 40K images of 256×256-pixel resolution in binary classification and recorded about 98% accuracy.

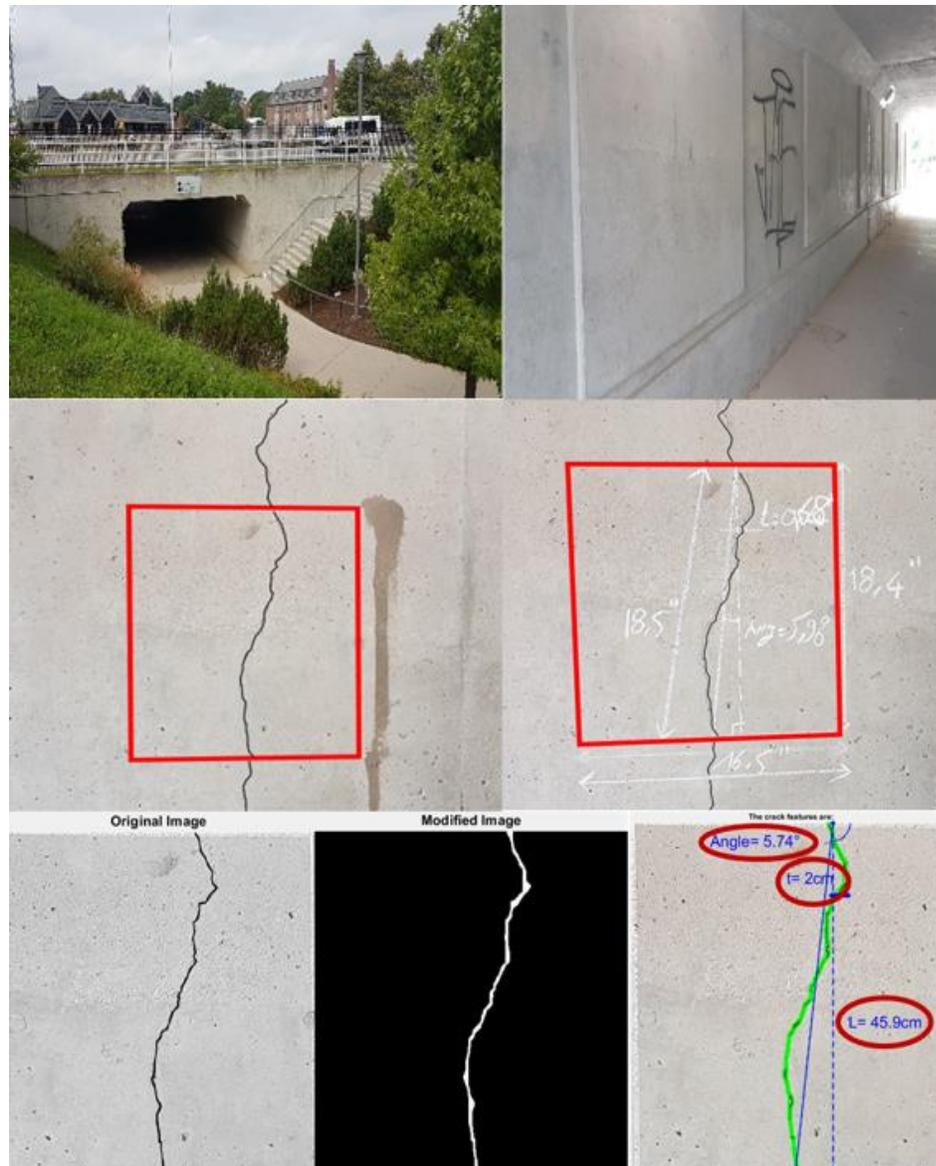


Figure 3.21: Calculation of Crack Features of a Damaged Reinforced Concrete wall.

Comparative studies were conducted to examine the performance of their proposed model with traditional IPTs (Sobel and canny edge detection). The results showed that CNNs presented better performance than that of conventional IPTs under realistic lighting conditions. (Dorafshan *et al.*, 2018) compared the performance of traditional edge detectors (Roberts, Prewitt, Sobel, Laplacian of Gaussian, Butterworth, and Gaussian) and AlexNet CNN. They found that CNN achieved the best performance of 86% compared to 53%-79% for other IPT edge detection techniques. Since CNNs outperformed IPTs in classification

tasks, CNNs were adopted in the present study, achieving classification accuracy of about 99% for a set of 6000 images of 227×227-pixel resolution for the binary classification (cracked or intact). This proves that the adopted CNN architecture performed even better than in the other two referenced above with more than a 10% increase in the accuracy and in less computing time.

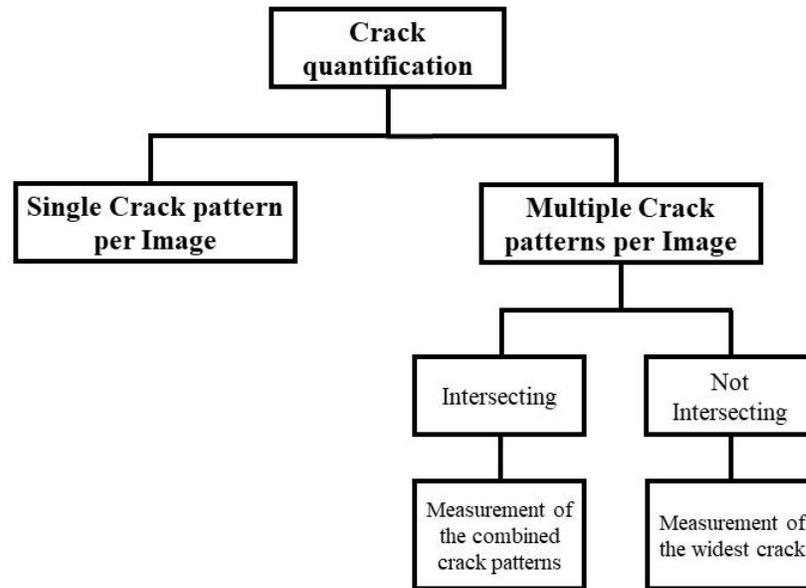


Figure 3.22: An Overview of the Crack Quantification Process.

However, this study is associated with some limitations. For instance, the model can, at this stage of development, accurately quantify one single crack pattern per Image. For a set of cracks (two or more patterns), the prediction results can be overestimated (more conservative) in terms of length, width, and angle of orientation. However, this risk is low, considering the small size of images. **Figure 3.22** explains the crack quantification process adopted in this study. If a single crack is present on the Image, then the model precisely quantifies its length, width, and angle of orientation, as presented in **Figures 3.20, 3.21, and 3.23a-b**. However, if multiple crack patterns are present on the Image and not intersecting at a certain point, as shown in **Figures 3.19 and 3.23c-d**, then only the most critical one (widest) is localized, segmented, and quantified. So, when having, for example, three crack patterns, one 3 mm, 0.1 mm, and 1 cm wide, the model finds its calculation and identify the degree of severity of damage based on the 1 cm crack (the widest).

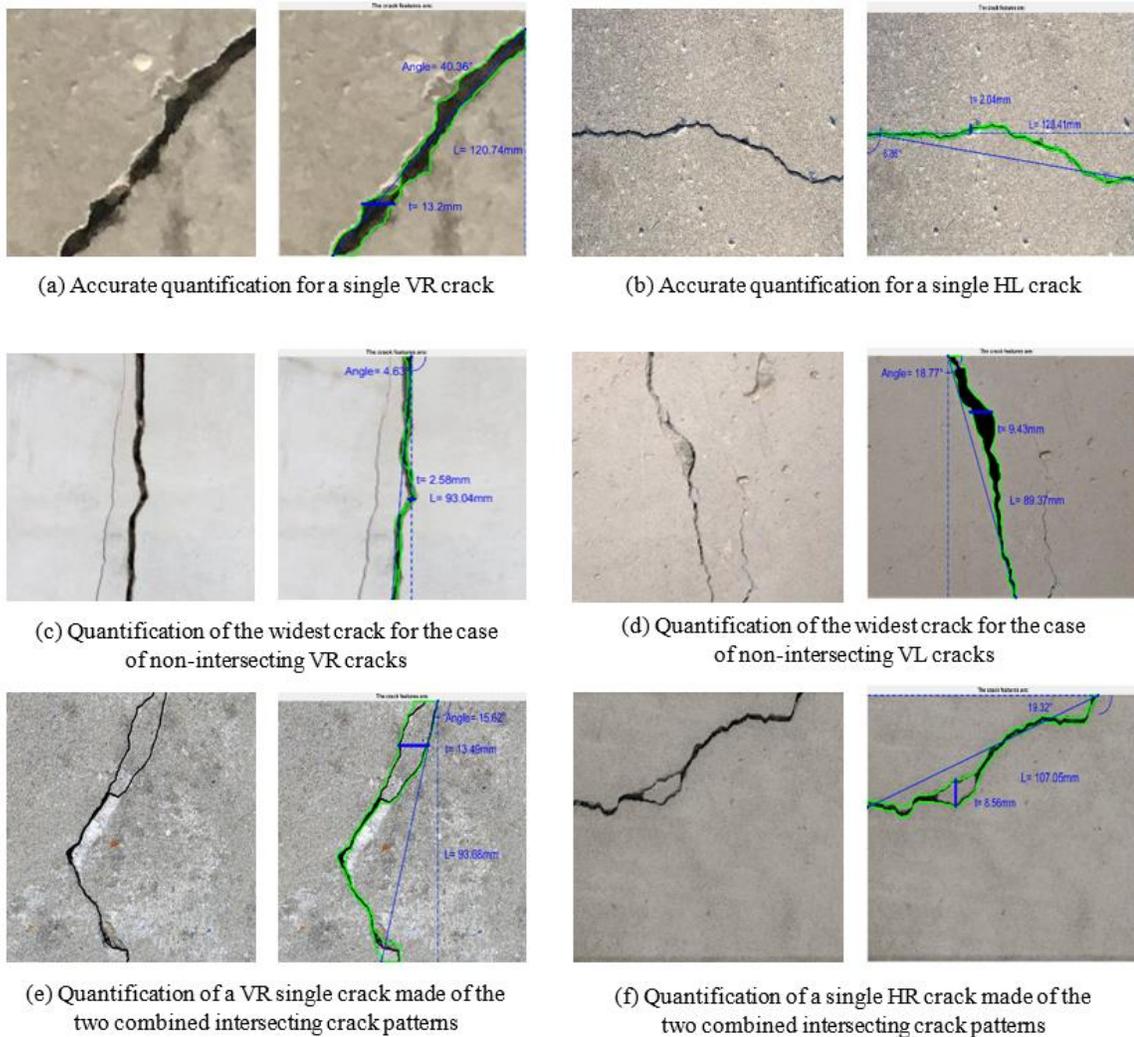


Figure 3.23: Performance of the Proposed Approach with a Questionable Explanation.

Therefore, crack quantification, damage type prediction, and its degree of severity are based on the widest crack. However, if the multiple cracks are intersecting, as shown in **Figures 3.18** and **3.23e-f**, then the model quantifies these kinds of crack as a single crack pattern based on their combination. This can be seen from the point of view that the part located between the two intersecting points in the Image along the way between the two intersecting cracks consists of a fully damaged area, and it is not transferring the full load in the corresponding structural element. Moreover, this model can detect at this stage of development crack widths and lengths starting from 0.1mm. Any crack dimension less than 0.1mm can not be well detected. Thus, this should be investigated in further studies.

Table 3.5: Comparison between experimental and numerical values for 100 epochs calculation.

| <i>Fig</i> | <i>M</i> | <i>O</i> | <i>Area</i> (<i>m</i> ²) | <i>L</i> (<i>cm</i>) | <i>L_ε</i> (%) | <i>W</i> (<i>cm</i>) | <i>W_ε</i> (%) | <i>Ang</i> (°) | <i>Ang_ε</i> (%) |
|--------------|---|----------|--|---------------------------|-------------------|---------------------------|-------------------|-------------------|---------------------|
| Figure 3.18 | E | VR | 0.0686 | 27.94 | 2.72 | 10.16 | 1.57 | 38 | 2.08 |
| | A | VR | | 28.7 | | 10 | | 38.79 | |
| Figure 3.19 | E | VL | 0.058 | 26.03 | 0.65 | 1.1 | 9.09 | 9 | 1.22 |
| | A | VL | | 26.2 | | 1 | | 9.11 | |
| Figure 3.20 | E | HR | 0.38 | 69.85 | 0.07 | 2.85 | 5.26 | 14.53 | 0.61 |
| | A | HR | | 69.9 | | 3 | | 14.44 | |
| Figure 3.21 | E | VR | 0.196 | 46.9 | 2.14 | 2.09 | 4.3 | 5.96 | 3.7 |
| | A | VR | | 45.9 | | 2 | | 5.74 | |
| Relative Avg | For the current set of four images presented in Figure 3.18, 3.19, 3.20, and 3.21 | | | | 1.39 | | 5.05 | | 1.9 |
| Global Avg | For the total number of the tested Images, 10 Images | | | | ~1.5 | | ~5 | | ~2 |

3.7 Conclusions

In this study, an automated inspection model for concrete structures using DL and IPTs to detect cracks is proposed. A convolutional neural network was trained independently on an image database consisting of 40k images with a 227x227 pixel resolution. The used classifier englobed five classes based on two criteria: the condition of the concrete surface (presence of cracks), and the orientation of cracks (HR, HL, VR, and VL). The total number of images used for training and testing of the classifier was 6000, with a split of 60:20:20 (3600 Images for training and 1200 images for validation and testing). IPTs have been implemented to induce transformations in the pictures tested by the CNN classifier for localizing cracks using Otsu's method for thresholding, noise removal, image binarization, and image segmentation. After localizing the crack, its geometrical properties, including length, width, and angle of orientation, are calculated. A field test was conducted to

evaluate the performance of the trained IPT algorithm by testing images of higher pixel resolution. Based on the experimental findings, the following conclusions can be drawn:

1. Convolutional Neural Networks, coupled with improved Otsu Image processing, can offer a powerful tool for classification, localization, segmentation, and quantification of damage in cracked cement-based materials and concrete structural elements.
2. Based on the classification analysis, the recorded testing accuracy was 98.25%, 97.18%, and 96.17% for the first, second, and third classifiers, respectively.
3. Based on the quantification analysis, the measurement error was 1.5%, 5%, and 2% for calculating the crack length, crack width, and crack angle of orientation.
4. The type of structural damage or durability related damage (e.g., corrosion) and its degree of severity were determined using different international standards and codes for buildings and bridges.
5. The damage detection, classification, and measurement method proposed in this study demonstrated superior performance and accuracy while requiring excellent computational and time efficiency compared to other existing methods.

3.8 References

- ACI Committee 224. "Control of cracking in concrete structures (ACI 224R-19)." (2019).
- Akbar, M. A., Qidwai, U., & Jahanshahi, M. R. (2019). An evaluation of image-based structural health monitoring using an integrated unmanned aerial vehicle platform. *Structural Control and Health Monitoring*, 26(1), e2276.
- American Association of State Highway and Transportation Officials, AASHTO LRFD Bridge Design Specifications, 2007.
- Antunes, P., Travanca, R., Rodrigues, H., Melo, J., Jara, J., Varum, H., & André, P. (2012). Dynamic structural health monitoring of slender structures using optical sensors. *Sensors*, 12(5), 6629-6644.
- Beckman, G. H., Polyzois, D., & Cha, Y.-J. (2019). DL-based automatic volumetric damage quantification using a depth camera. *Automation in Construction*, 99, 114-124.
- Canadian Standards Association, Canadian High Way Bridge Design Code, CAN/CSA-A23.3, Mississauga, Ontario, Canada. (2004).

- Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G.-B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating DL Applications. *IEEE Access*, 6, 61677-61685.
- Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). DL-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5), 361-378.
- CSA-S16-14, 2017. Canadian Highway Bridge Design Code, Toronto, Canada.: Canadian Standards Association.
- Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, 186, 1031-1045.
- Folk, M., Heber, G., Koziol, Q., Pourmal, E., & Robinson, D. (2011). An overview of the HDF5 technology suite and its applications. In *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases* (pp. 36-47). ACM.
- Gao, Y., & Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, 33(9), 748-768.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *DL*: MIT press.
- Hoang, Nhat-Duc. "Detection of surface crack in building structures using an image processing technique with an improved Otsu method for image thresholding." *Advances in Civil Engineering 2018* (2018).
- Hong, H. (2000). Assessment of reliability of aging reinforced concrete structures. *Journal of Structural Engineering*, 126(12), 1458-1465.
- Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167* (2015).
- JSCE (Japan Society of Civil Engineers). "Standard Specifications for concrete structures -2007" Tokyo, Japan: JSCE, (2010).
- Kannojia, Suresh Prasad, and Gaurav Jaiswal. "Effects of Varying Resolution on Performance of CNN based Image Classification: An Experimental Study." *International Journal of Computer Sciences and Engineering* 6.9 (2018): 451-56p.
- Kim, H., Ahn, E., Shin, M., & Sim, S. H. (2019). Crack and non crack classification from concrete surface images using machine learning. *Structural Health Monitoring*, 18(3), 725-738.
- Kim, I. H., Jeon, H., Baek, S. C., Hong, W. H., & Jung, H. J. (2018). Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle. *Sensors*, 18(6), 1881.

- Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., & Fieguth, P. (2015). A review of computer vision-based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, 29(2), 196-210.
- Kohavi, R., & Provost, F. (1998). Confusion matrix. *Machine learning*, 30(2-3), 271-274.
- Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Y. J. Z., Zhang, L., Yang, F., Zhang, Y. D. & Zhu, Y. J. (2016). Road Crack Detection Using Deep Convolutional Neural Network. In 2016 IEEE International Conference on Image Processing (ICIP). <http://doi.org/10.1109/ICIP.2016.7533052>
- Maguire, M. D., Sattar, and Thomas, Robert J. (2018). "*SDNET2018: A concrete crack image dataset for machine learning applications*". Retrieved from https://digitalcommons.usu.edu/all_datasets/48
- Mosley, William Henry, Ray Hulse, and John Henry Bungey. Reinforced concrete design: to Eurocode 2. Macmillan International Higher Education, 2012.
- New Zealand Standard. NZ3101.1:2006 Concrete structures standard.
- Nowak, A. S. (2012). Bridge evaluation, repair, and rehabilitation (Vol. 187): Springer Science & Business Media.
- Özgenel, Ç. F. (2019). "*Concrete Crack Images for Classification*." Retrieved from <http://dx.doi.org/10.17632/5y9wdsg2zt.2>
- Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. *Pattern recognition*, 26(9), 1277-1294.
- Richardson, Mark G. Fundamentals of durable reinforced concrete. CRC Press, 2002.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Song, G., Gu, H., Mo, Y., Hsu, T., & Dhonde, H. (2007). Concrete structural health monitoring using embedded piezoceramic transducers. *Smart Materials and Structures*, 16(4), 959.
- Sun, Y., Salari, E., & Chou, E. (2009). *Automated pavement distress detection using advanced IPs*. Paper presented at the 2009 IEEE International Conference on Electro/Information Technology.
- Standards Australia. AS 3600:2018 Concrete Structures - Concrete Institute of Australia.
- Standards Australia. AS 5100.5-2004 Bridge design – Concrete.
- Xu, Yang, et al. "Automatic seismic damage identification of reinforced concrete columns from images by a region-based deep convolutional neural network." *Structural Control and Health Monitoring* 26.3 (2019): e2313.
- Ye, Xiao-Wei, Tao Jin, and Peng-Yu Chen. "Structural crack detection using deep learning-based, fully convolutional networks." *Advances in Structural Engineering* 22.16 (2019): 3412-3419.

Yu, S.-N., Jang, J.-H., & Han, C.-S. (2007). Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. *Automation in Construction*, 16(3), 255-261. doi:10.1016/j.autcon.2006.05.003.

Yuheng, S., & Hao, Y. (2017). Image segmentation algorithms overview. *arXiv preprint arXiv:1707.02051*.

Chapter 4

4 Localization and Classification of Structural Damage using 1D CNN Single-Channel Signal-Based Measurement

Rapid and diligent identification of structural integrity risk has been a core thrust of SHM systems. Vibration-based SHM and damage detection have gained paramount importance over the last two decades. More recently, substantial research efforts have been devoted to DL algorithms, which yielded accuracy unmatched by existing conventional approaches. The present chapter proposes a novel DL-based damage detection approach to extract features from raw acceleration sensor data automatically. A new One-Dimensional Convolutional Neural Network (1D CNN) named *BuildingNet* was designed to learn features and identify damage locations in real-time under different damage assessment scenarios. Parametric studies were conducted on different layer numbers, size of training datasets, and noise levels. An ensemble of systematic studies on the optimization of network architecture and preparation of the training data was performed. Numerical investigations on a mid-rise building were conducted to demonstrate the accuracy and efficiency of the proposed model framework compared with traditional ML methods. Time-domain monitoring data, both from multiple and single-channel measurements, were used for training and testing three different architectures for *BuildingNet*.

4.1 Introduction

Civil structures and infrastructures are often subjected to numerous external loads. Such loads are caused by many factors such as earthquakes, winds, changes in temperature and moisture gradients, chemical attack, and excitation produced by humans. During the service life of these structures, loads can result in structural damage or cause catastrophic failure, with associated economic and loss of human life. Damage is generally defined as a reduction in local stiffness. It can be due to cracks, spalling, and even total failure of structural members such as beams, columns, and member connections (Farrar *et al.*, 2007). Structural damage identification has gained increasing attention of the engineering community to maintain the structural performance of civil assets and better manage limited repair and rehabilitation resources. This process is often done via techniques that can assess damage objectively and adequately at the earliest possible time. This should allow

predicting the remaining useful life of assets to mitigate failures and optimize resource allocation.

Traditional damage assessment tools that depend on visual inspection have often proved inefficient since they involve safety risks, suffer from limited access, and tend to be laborious and time-consuming. They are also subjective since different inspectors could render diverging appraisals. To accurately assess structural damage, more reliable and effective non-destructive damage identification techniques are indispensable. Such methods can generally be categorized either as local or global (Doebbling *et al.*, 1996). Local techniques, including ultrasonic and X-ray methods, require prior knowledge of the vicinity of damage and easy access, which is not always guaranteed. Thus, vibration-based damage identification methods have emerged as global damage identification techniques to overcome these difficulties. These methods aim to assess the presence, severity, and location of defective areas by processing signals measured via a network of sensors.

A broad range of vibration-based techniques, algorithms, and methods was developed to solve various problems encountered in-situ. They can be classified into parametric (model-based) and nonparametric (signal-based) techniques (Abdeljaber *et al.*, 2017). Parametric methods employ identification algorithms to the measured response to determine damage sensitive features, including natural frequencies, modal damping, and mode shapes that affect the current or future performance of a structure caused by an alteration of the physical properties (mass, damping, and stiffness). Alterations to the parameters prevailing in the healthy structural state are used to identify structural damage. Conversely, nonparametric methods use statistical means to recognize damage directly from measured signals.

Recently, increasingly research efforts have applied ML, encompassing a broad range of parametric and nonparametric studies. Such ML algorithms proved efficient in dealing with this since they can learn complex non-linear relationships between acquired signals and the state of the structural system (Abellan-Nebot *et al.*, 2010). Most ML-based damage detection methods involve two main steps: i) feature extraction, and ii) feature classification. For instance, SVM was one of the first proposed ML techniques to extract damage features from signals. This algorithm was successful in performing small samples

of classification and regression (Suykens *et al.*, 1999). Additionally, Artificial Neural Networks (ANN) have been extensively used to fulfill the same task since they have robust pattern recognition and classification capabilities owing to their self-learning function, the capacity of storage, and the ability of rapid search for optimal solutions (Zhang *et al.*, 1998).

However, the process of damage extraction using such techniques needs advanced algorithms and computational resources to accurately extract the main damage features from the acquired signals and limit the manual process of feature extraction. Thus, research efforts in ML algorithms are still pursuing higher performance and efficiency. Until now, there is no specific answer to which feature extraction method can optimally characterize the acquired signal. Moreover, this process is costly and time-consuming, which has compromised its real-time application.

To overcome such limitations, numerous DL algorithms have been assigned to this nonparametric time-series problem. One of the solutions consisted of capturing the representation information adaptively and fusing the feature extraction and feature classification in a combined optimization process. DL has reached great success in image recognition problems dealing with SHM of civil infrastructure. For instance, a DL method for image based SHM was proposed, and it surpassed traditional image processing methods in finding cracks under realistic conditions (Cha *et al.*, 2017). Other studies on the application of DL to image SHM applications were reported in the literature (Zhang *et al.*, 2020) (Xu *et al.*, 2019) (Kim *et al.*, 2019). However, such techniques face some challenges, such as data preprocessing, which is essential for automatic real-time monitoring and alarming of SHM systems and data-based off-line long-term performance analysis of structures.

Data preprocessing prepares the data and make it useful for the DL training process. This can be done by transforming the acceleration signals into spectrograms (2D Images) since DL achieved great success in dealing with images. Subsequently, based on these spectrograms, DL algorithms such as Two Dimensional Convolutional Neural Networks (2D CNNs) are applied to classify structures as damaged or not. For instance, a CNN based approach was used to classify and predict various types of delamination in composite laminates using low-output structural vibrations (Khan *et al.*, 2019). This was done using

the Short-Time Fourier Transform (STFT) to get a 2D spectral frame representation from the transient responses. Another study had implemented a 2D CNN to assess and localize damage in a long-span cable-stayed bridge. The measured acceleration containing six patterns of data anomaly was first transformed into a grayscale image used as a training set for the network (Bao *et al.*, 2019). Another implementation of 2D CNNs was reported, where the measured vibration response was represented as matrices and fed to the CNN as input images. Their proposed method's efficacy was validated using a limited number of vibration response data recorded during the shake-table testing of a one-fourth-scale model of a highway bridge (Khodabandehlou *et al.*, 2019). A further study proposed an approach that consisted of localizing the damage for building structures using dynamic displacement responses based on a 2D CNN. Based on the interrelation constructed by CNN in advance, damaged stories are localized by investigating the discrepancy of dynamic responses between healthy and damaged states (Oh *et al.*, 2020).

However, there are certain drawbacks and limitations in using these deep architectures (2D CNN). For instance, despite the relatively good performance, the studies reported that the training phase was computationally time-consuming. They also pose problems with high computational complexity, requiring specialized training hardware. Moreover, they are not suitable for online damage detection. Training this kind of CNNs also needs a massive amount of dataset to achieve good classification results, which does not apply to the case of signals because of their scarcity (Kiranyaz *et al.*, 2019). Therefore, other DL techniques, such as Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), and 1D CNN, have been proposed to process data automatically, without the requirement for passing by vibration images.

These techniques proved efficient in the case of a network of sensors. For instance, a 1D CNN was used to track damage in a grandstand simulator, having 30 accelerometers installed on 30 joints (Abdeljaber *et al.*, 2017). Another application had conducted several impact-hammer tests on a steel frame for six installed accelerometers in a different position (Zhang *et al.*, 2019). Generally, such applications were applied to horizontal structures (e.g., grandstand simulator of 1.65 m height and a steel frame of 1.5 m height) typically

consisting of continuous acquisition of signals by a network of accelerometers (multiple channel measurements).

Therefore, concerted DL research efforts are needed to monitor damage into vertical structures (e.g., multi-story buildings). Hence, in the present study, a DL comparative analysis of signal processing techniques for vibration based civil SHM is proposed. The study explores the performance of DL models, including 1D CNN, Long Short-Term Memory networks, and Resnet18, and other conventional ML algorithms. The comparison is based on different damage scenarios for a Ten Degree of Freedom building system (10 DOF) and different disposition of sensors (e.g., Multi-channel measurement: one sensor for each story and one single-channel measurement: one sensor for the entire building).

The novelty of the present study consists of the following aspects: i) Developing a real-time DL vibration-based technique for assessing damage in mid-rise buildings with high accuracy and short computational time by automating the damage sensitive features extraction pre- and post-processing; ii) Proposing a single-channel measurement (only one sensor) vibration-based damage detection platform that can detect and assess the health of a structure, which is more economical and practical; and iii) Demonstrating the performance and robustness of the proposed model by adding up to 20% random Gaussian noise.

The chapter is organized as follows: Section two details the methodology adopted in this study including the DL network architectures; Section three describes the data preparation scheme; Section four defines the experimental study, presents the results and performance evaluation of the DL models, and delivers an in-depth discussion summarizing the findings of the approach; Finally, Section five draws the main conclusions and suggests potential directions of future research.

4.1.1 Research Significance

This work coins a novel 1D CNN called *BuildingNet* adopted for mid-rise buildings to bridge research gaps identified above. A state-of-the-art network configuration was proposed, and appropriate changes were made in this study to fit the scenarios proposed.

Different influential factors have been successfully tested, including the number of convolution blocks, the effect of data increase, and the robustness of the algorithm against 20% random Gaussian noise. The proposed model was also compared to other conventional ML and DL models. The findings should encourage a critical look into the state-of-the-art of DL single-channel measurement vibration-based structural damage evaluation protocols and highlight the need for comprehensive studies that can assist engineers in carrying out a robust inspection of civil engineering structures.

4.2 Methodology

This section describes the architecture of the designed CNN and introduces the function and background of each of its layers. A typical CNN contains at least one convolutional layer, a max-pooling layer, a flatten layer, a connected layer, and a softmax output layer. The corresponding number of hidden layers can be determined according to the complexity of the target problem. Convolutional and flatten layers are usually followed by batch normalization and a dropout layer, respectively, which boosts the performance of CNN.

4.2.1 1D CNN Network Architecture

In this study, three different 1D CNN architectures were implemented to assess their performance in dealing with the problem of structural damage identification under different damage assessment scenarios. As illustrated in **Figure 4.1**, the first adopted scenario is "BuildingNet_1", which is expected to yield high classification accuracy, corresponding to the case of having one installed sensor on each floor (case study 1). This proposed model aims to detect not only the damage state of the building (healthy or damaged), but also localize the exact location of damage (in which floor). The second proposed architecture is "BuildingNet_2", which corresponds to the scenario of having only one accelerometer installed on the first floor of the building in an attempt to propose a DL model that can detect and localize damage separately on each story using a single-channel measurement (case study 2). The last architecture, "BuildingNet_3" corresponds to the case of only having one sensor to classify the global state (healthy or damaged), without determining the exact floor where damage is located (case study 3). Since the number of sensors and prediction categories is tailored to a given assessment scenario, the scenario dictates the

shape of the input and output layers. **Table 4.1** presents the main layers of each of the three proposed architectures.

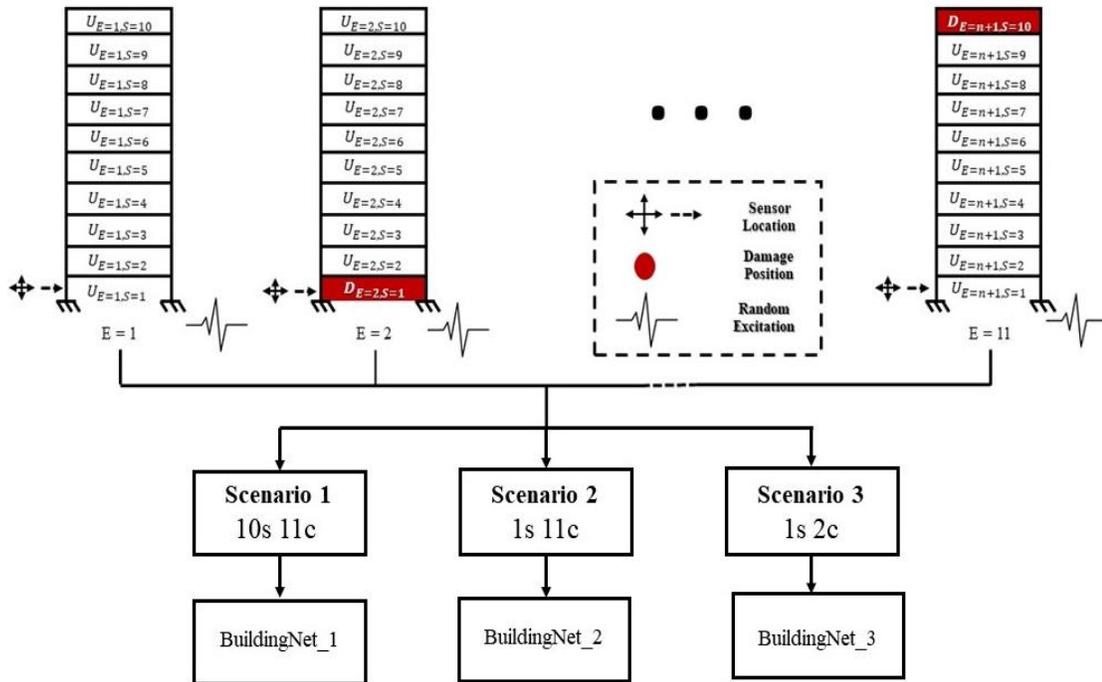


Figure 4.1: BuildingNet Design Methodology.

4.2.2 CNN Representation

4.2.2.1 1D CNN layer

1D CNNs are a subset of convolutional neural networks. The application of CNNs was first developed for image classification, having great success in dealing with problems in which the DL model receives 2-dimensional input representing the pixels and color channels of an image. This is known as feature learning and was applied similarly to one-dimensional sequences of data. The model excerpts feature from sequences of data and outputs the internal features of the sequence. 1D CNN proved their effectiveness in dealing with time-series sensor data, signal data analysis over a fixed length period (audio recording), and Natural Language Processing (NLP). The main difference between 2D CNN and 1D CNN is the structure of the input data and how the filter moves across the data (**Figure 4.2**). Another difference is that 1D networks use larger filter sizes. In a 1D network, a kernel of

size f contains only $f1$ feature vectors, whereas, in a 2D CNN, a filter of size $f2$ contains $f2 \times f2$ feature vectors, making this selection very broad.

Figure 4.3 illustrates two types of layers for a 1D CNN. The first is the CNN-layers, where the convolution and pooling operations co-occur; the second is the fully connected layers or the Multi-Layer Perceptron (MLP). The configuration of a 1D CNN is made of 1) Number of hidden CNN layers and MLP layer/neurons, 2) Filter size in each CNN layer, 3) Subsampling factors for each CNN layer and 4) Pooling and activation functions. **Figure 4.4** presents three consecutive CNN layers of a 1D CNN. In each CNN-layer, a 1D forward propagation (FP) from the previous convolution layer ($l-1$) to the current layer's input neuron (l), is expressed in **Equation 4.1**.

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} conv1D(w_{ik}^{l-1}, s_i^{l-1}) \quad (4.1)$$

Where x_k^l is the input, b_k^l is the bias of the k^{th} neuron at layer l , and s_i^{l-1} is the i^{th} neuron's output at layer $l-1$. The filter weight from the i^{th} neuron at layer $l-1$ to the k^{th} neuron at layer l is noted as w_{ik}^{l-1} . The intermediate output of the neuron, y_k^l is then expressed using the input x_k^l , as in **Equation 4.2**.

$$y_k^l = f(x_k^l) \text{ and } s_k^l = y_k^l \downarrow ss \quad (4.2)$$

Where s_k^l is the neuron's output, and $\downarrow ss$ is the down-sampling operation with the factor, ss . More details on the training methodology for the 1D CNN are presented in a well-explained study (Kiranyaz *et al.*, 2015).

4.2.2.2 Batch Normalization (B.N.) Layer

The training data are learned batch by batch. As a result, the batch distributions are non-uniformly and unstably distributed and must be fitted by the network parameters in every training iteration, which significantly shows the convergence of the model. To tackle this problem, the convolutional layer is followed by an adaptive reparameterization method called batch normalization (Ioffe *et al.*, 2015).

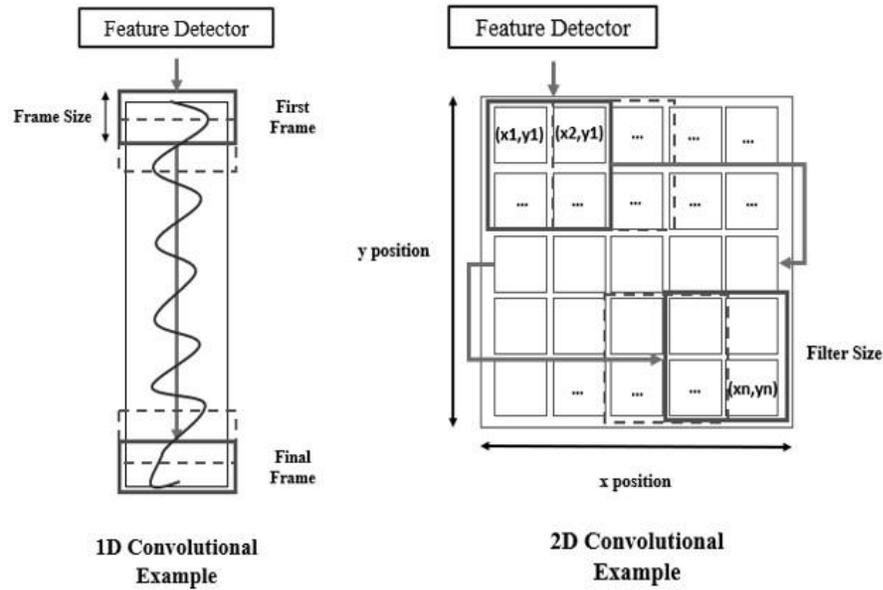


Figure 4.2: Difference between 1D CNN and 2D CNN.

The batch normalization algorithm calculates the mean and variance of every training data batch, then shifts and scales the original data to zero-mean and unity variance. Afterward, the shifted data is assigned to weight and bias to increase their effect. This reparameterization using BN helps to alleviate the problem of coordinating updates through the layers in the neural network.

4.2.2.3 Rectified Linear unit (ReLU)

The activation function is used in a neural network to transform the summed weighted input from the node into its activation (Agarap *et al.*, 2018). The rectified linear activation function is a linear function that outputs the input directly if it is positive. Otherwise, it outputs zero. It has become the standard activation function for many types of neural networks owing to its performance and simplicity for training purposes. A non-linear ReLU function was used as an activation function, as shown in **Equation 4.3**.

$$f(x) = \max(0, x) \quad (4.3)$$

Table 4.1: Adopted 1D CNN Architectures.

| Network Architecture | Feature Extraction | Classification |
|-----------------------------|--|---|
| BuildingNet_1 | Sequential ((0): Conv1d (1, 8, kernel_size = (11,), stride = (1,), padding = (1,), dilation = (1,)) (1): BatchNorm1d (8) (2): ReLU () (3): Conv1d (8, 8, kernel_size = (3,), stride = (2,), padding = (1,), dilation = (1,)) (4): BatchNorm1d (8) (5): ReLU () (6): Conv1d (8, 8, kernel_size = (3,), stride = (2,), padding = (1,), dilation = (1,)) (7): BatchNorm1d (8) (8): ReLU () (9): Conv1d (8, 8, kernel_size = (3,), stride = (2,), padding = (1,), dilation = (1,)) (10): BatchNorm1d (8) (11): ReLU () (12): Conv1d (8, 8, kernel_size = (3,), stride = (2,), padding = (1,), dilation = (1,)) (13): BatchNorm1d (8) (14): ReLU () (15): Conv1d (8, 8, kernel_size = (3,), stride = (2,), padding = (1,), dilation = (1,)) (16): Flatten () (17): Linear (in_features = 256, out_features = 256, bias = True) | Sequential ((0): Linear (in_features = 256, out_features = 256, bias = True) (1): ReLU () (2): Dropout (0.5) (3): Linear (in_features = 256, out_features = 128, bias = True) (4): ReLU () (5): Dropout (0.5) (6): Linear (in_features = 128, out_features = 11, bias = True) |
| BuildingNet_2 | Sequential ((0): Conv1d (1, 8, kernel_size = (11,), stride = (1,), padding = (1,), dilation = (1,)) (1): BatchNorm1d (8) (2): ReLU () (3): Conv1d (8, 8, kernel_size = (3,), stride = (2,), padding = (1,), dilation = (1,)) (4): Flatten () (5): Linear (in_features = 4064, out_features = 256, bias = True) | Sequential ((0): Dropout (0.5) (1): Linear (in_features = 256, out_features = 128, bias = True) (2): ReLU () (3): Dropout (0.5) (4): Linear (in_features = 128, out_features = 11, bias = True) |
| BuildingNet_3 | Sequential ((0): Conv1d (1, 8, kernel_size = (11,), stride = (1,), padding = (1,), dilation = (1,)) (1): BatchNorm1d (8) (2): ReLU () (3): Conv1d (8, 8, kernel_size = (3,), stride = (2,), padding = (1,), dilation = (1,)) (4): BatchNorm1d (8) (5): ReLU () (6): Flatten () (7): Linear (in_features = 4064, out_features = 256, bias = True) | Sequential ((0): Linear (in_features = 256, out_features = 256, bias = True) (1): ReLU () (2): Dropout (0.5) (3): Linear (in_features = 256, out_features = 128, bias = True) (4): ReLU () (5): Dropout (0.5) (6): Linear (in_features = 128, out_features = 2, bias = True) |

4.2.2.4 Dropout Layer

Dropout is one of the techniques used to tackle the problem of overfitting for neural networks and to generate much more efficient training examples by reducing the coadaptation between neurons. Overfitting is determined whenever the training loss is much smaller than the testing loss. The core idea behind dropout is to randomly disconnect the connections between neurons, having a fixed dropout rate. A dropout coefficient of 0.5 was used in the present study.

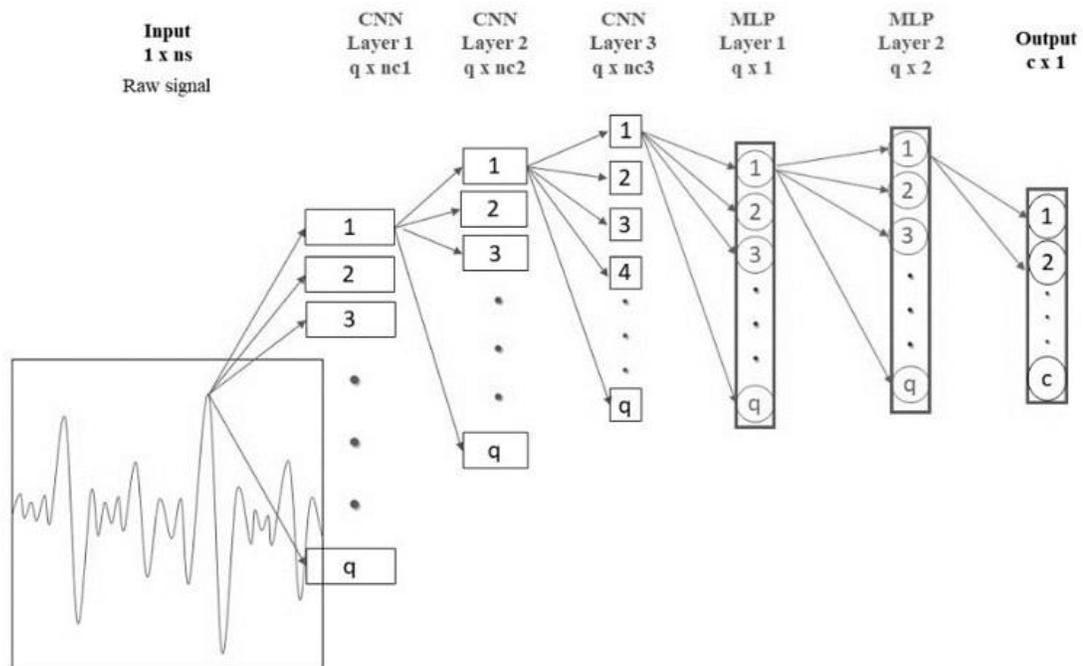


Figure 4.3: Example of 1D CNN Configuration with 3 CNN and 2 MLP Layers.

(reproduced and modified from Kiranyaz et al., 2019)

4.2.2.5 Fully Connected (F.C.) or Linear Layers

A fully connected layer is a linear layer that applies a linear transformation to the incoming data. It takes the outputs of the previous layers, flattens them, and transforms them into a single vector that can input to the next phase, following **Equation 4.4**, where A is the input, b is the bias, and y is the output. If $\text{bias} = \text{False}$, then $b = 0$.

$$y = xA^T + b \quad (4.4)$$

4.2.2.6 Structural State Identification

1D CNN learns to identify the damage state by performing two other steps. The first step is to train the 1D CNN, and the second is to test it. In this study, the adopted training to testing ratio was 90:10. This ratio was evaluated against other rates (80:20 and 70:30). Nevertheless, the proposed ratio is considered the best in terms of performance metrics. In the training step, the 1D CNN is updated by evaluating and reducing the deviations between the predicted 1D CNN output and actual labeled locations of the local structural-state changes, on a batch-by-batch basis. **Figure 4.5** presents the main architecture parameters corresponding to three case studies of *BuildingNet*.

Table 4.2 presents the corresponding hyper-parameters for the three 1D CNN architectures adopted in this study. For instance, the training batch size was set to 256. The uniformity and deviations between the network outputs and labels are called the accuracy and loss, respectively. The loss of the CNN is determined by the categorical cross-entropy loss function, computed by **Equation 4.5**, where E_l is the loss, N_t is the number of training data, x denotes the samples, y means the labels, and a are the predictions (outputs of the neural network).

$$E_l = -\frac{1}{N_t} \sum_x [y \ln a + (1 - y) \ln(1 - a)] \quad (4.5)$$

Table 4.2: Adopted 1D CNN Hyperparameters.

| | BuildingNet_1 | BuildingNet_2 | BuildingNet_3 |
|----------------------------|---------------------------------|----------------------|----------------------|
| Learning rate | 0.0001 | 0.001 | 0.0001 |
| Training batch size | 256 | | |
| Testing batch size | 10 | 5 | 5 |
| Epochs | 1000 | | |
| Loss function | Categorical Cross-Entropy (CCE) | | |
| Optimizer | Adam | | |

To reduce the loss of the proposed models, the present study uses Adam optimizer, which is an adaptive learning-rate optimization algorithm, to update the 1D CNN parameters. The proposed optimizer computes the gradients of the 1D CNN parameters and updates them in every iteration. The used learning rates are different for the three proposed models due to the complexity of data and damage assessment scenarios.

Grid search Hyperparameters optimization was used to find the best hyperparameters including the number of epochs (1000), learning rate (e^{-3} or e^{-4}), testing batch size (10 or 5), and the optimizer (Adam). The 1D CNN is trained by iteratively feeding the training data, evaluating the loss, and updating the 1D CNN parameters. The training is complete when the CNN outputs achieve high accuracy and $f1$ score with a relatively low loss.

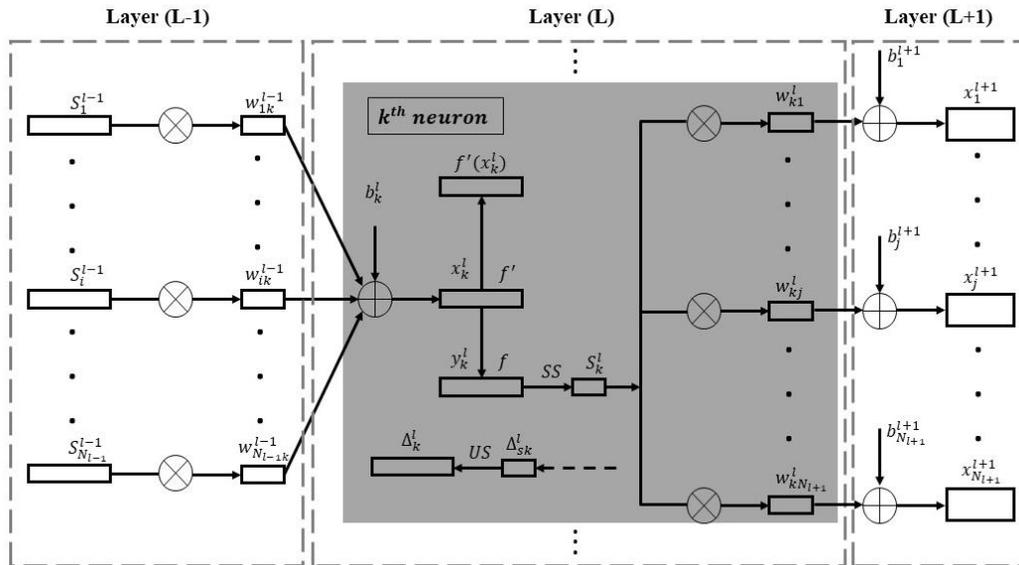


Figure 4.4: Three 1D CNN Consecutive Hidden CNN Layers. (reproduced and modified from Abdeljaber et al., 2017)

4.3 Data Preparation

4.3.1 Data Description

4.3.1.1 Data Features

The vibration test data in the time-domain is obtained following two procedures. The first consists of getting the training data from N_s accelerometers, where N_s corresponds to the

number of stories (10 sensors for 10 DOF). The second approach obtains the data from a single accelerometer installed in the first story of the building. Ideally, all possible combinations of structural damage scenarios need to be covered. Nevertheless, this is nearly impossible since their number is infinite, requiring vast computational resources. In this study, the classes of the training data are designed according to the damage location.

The data used for training the DL algorithms are obtained under different damage assessment scenarios. The first case study, "BuildingNet_1," corresponds to ten installed sensors per building to detect the damage on each floor separately. The second case study, "BuildingNet_2," tracks the damage on each level using one single-channel measurement. The third case study, "BuildingNet_3," detects the damage of the building in binary classification to decide whether the building is healthy or damaged. To select the appropriate test method for this study, a random shake excitation is adopted, since it is the most representative for real structures under different kinds of loads (e.g., traffic, environmental, seismic, wind). The data corresponds to the raw acceleration signals without applying any filtering (Abdeljaber and Avci, 2016), unlike conventional frequency domain (FD) methods (Pathirage *et al.*, 2018). Time-domain data contains all information, including the non-linear and transient effects, which can be missed by FD methods.

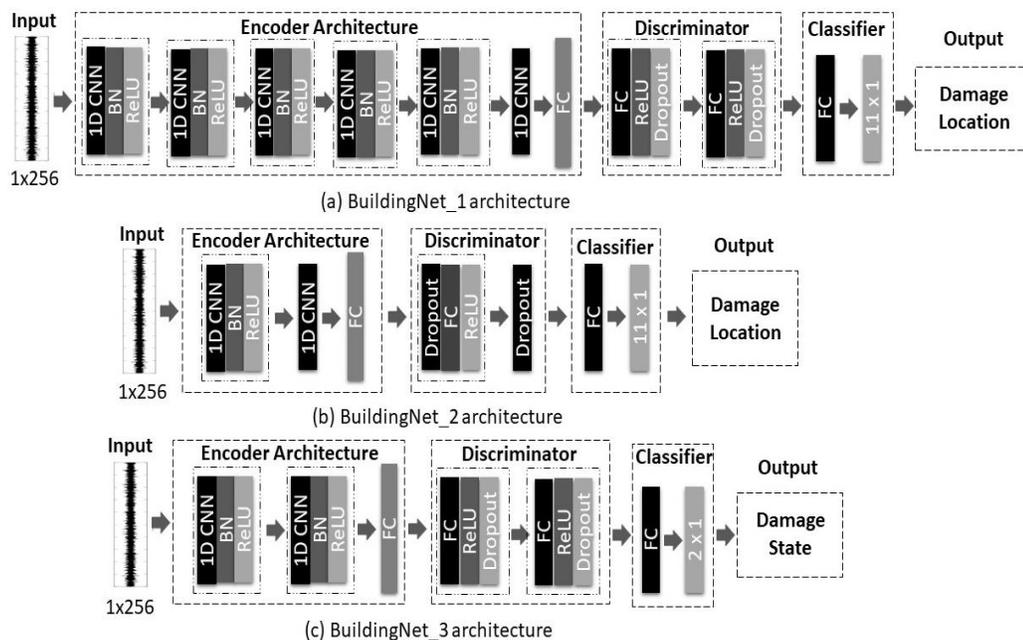


Figure 4.5: *BuildingNet* Corresponding to the Three Adopted Case Studies.

4.3.2 Data Decomposition using the sliding window technique

The sliding window is a signal decomposition technique used to reframe the time series data in order to apply supervised ML and DL models. Given a sequence of numbers for the time series dataset, the data can be restructured to several frames of time series data. This is done using the previous time steps as input variables and the next time steps as output variables. The number of previous time steps is called the window width. To reproduce more samples for the classification of these frames, the sliding window can overlap for a given time step, which is a tool that can be used for data augmentation (Kohavi and Provost, 1998). **Figure 4.6** demonstrates an example of signal decomposition into N_f frames using the sliding window technique. In this study, three different frame widths were tested (256, 512, and 1024). The last one yielded the best training performance (1024 samples), mainly including $N_f = 250$ frames. N_f can be obtained by using **Equation 4.6**.

$$N_f = \frac{1}{\text{Frame width}} \times \left(\frac{1}{\text{Time Step}} \times \text{Time record} \right) \quad (4.6)$$

4.3.3 Data Augmentation using SMOTE

As can be observed in data generation, there is a concern with unbalanced data. Data imbalance is defined as a difference in the number of data points for each class. This problem is especially persistent for the first and last models. This is because the undamaged vectors are more dominant, while damaged vectors are significantly less. For example, "BuildingNet_1" was made of 11 files, the first one corresponds to the healthy configuration of the building (no created damage) and ten other undamaged columns corresponding to 10 installed sensors at ten floors. Then, starting from the second file until the eleventh, there is only one column in each file that corresponds to the damaged pattern. For example, for the second file, which corresponds to the damage at the first story, its first column (for the 1st sensor installed at the first story) is the corresponding damaged vector. Thus, for these 11 files, one damaged vector for each class (e.g., D1: Damage in story 1) compared to 100 undamaged vectors for the healthy category. This results in a large order of difference, which is 100 times larger than the other classes. However, for "BuildingNet_3", this issue

is less prominent since the order of difference between the most abundant category (Healthy) and the other classes is ten times.

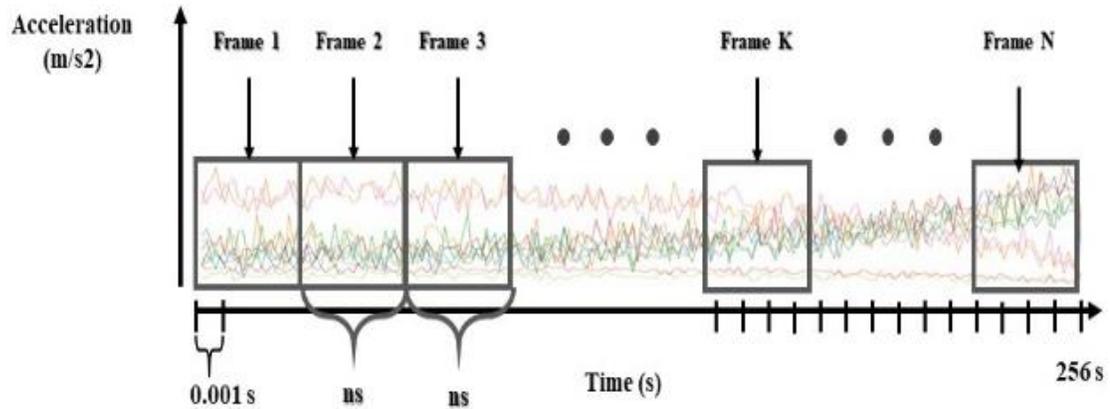


Figure 4.6: Example of a Signal Decomposition into N Number of Frames.

This difference is due to moving from ten sensors to only one for monitoring the damage on each floor separately. To tackle this problem, a method called Synthetic Minority Oversampling Technique (SMOTE) (Fernández *et al.*, 2018) was used to increase the number of cases in the dataset in a balanced way and oversample the minority class. The model works by generating new instances from existing minority classes. The new examples are a generation of new samples from the feature space of each target class and its nearest neighbors. Thus, SMOTE was adopted in "BuildingNet_1" and "BuildingNet_3" to oversample the damaged classes. This technique was only applied to the training set. However, the testing set should always be intact and original, and the performance of the model is tested on unmodified original and unseen data.

4.3.4 Data Processing

Training the DL models requires generating a dataset that consists of several undamaged/damaged acceleration signals for each story. Damage is created by reducing the equivalent stiffness that can be obtained based on a decrease of the moment of inertia. However, it should be noted that only the stiffness reduction is considered, and the mass change is ignored since structural damage is mainly related to stiffness reduction.

Since two approaches are adopted in this study, as illustrated in the previous paragraph, two data preparation schemes are presented. The first method consists of installing as many sensors as the number of floors. For each structure having a total of N_s stories (accelerometers), a unique CNN is assigned for the entire building, contrary to the study cited in³, where one single CNN for each sensor was implemented. To generate the training dataset needed to train each DL model, it is required to conduct $n + 1$ experiments. In the first experiment ($E = 1$), N_s acceleration signals are measured for the intact structure (Healthy state). The resulting signals are denoted as $U_{E=1,J=1}, U_{E=1,J=2}, \dots, U_{E=1,J=N_s}$. U denotes the signal measured at an undamaged story, E designates the experiment number, and J is the story number.

The remaining experiments are conducted one by one in sequential order. In each test $E = k+1$, the damage is introduced at the story $j = k$. The damage is induced by reducing the stiffness of each corresponding story, e.g., if $k1 = 1750$ N/m (No damage), after introducing 50% damage to the story, $k1$ will be equal to 875 N/m, and the n acceleration signals are measured under random excitation.

The measured signals are denoted as $U_{E=k+1,J=1}, \dots, D_{E=k+1,J=k}, \dots, U_{E=k+1,J=N_s}$, where D indicates that this signal was measured at the damaged story k . After performing the $n+1$ experiment, the signals that are measured at each story i are grouped together to create the damaged/undamaged vectors required to train the networks. The second method obtains the acceleration signals from one single channel measurement (one single sensor available at a time for the whole building). The second approach is created in pursuit of exploring damage identification using a decentralized sensing approach where one sensor is autonomously moved from one location to another.

4.4 Results and Discussions

This section describes the damage characterization performance results that are obtained when the proposed framework was applied to the numerical data for the three 10 DOF models and the different conducted experiments. As explained in Section 2, three damage assessment scenarios were implemented in this study. The first scenario is the least complicated in terms of damage monitoring. It consists of installing as many sensors as the

number of floors of the 10 DOF system. However, this scenario is expected to be very costly. It is also impractical to implement sensors on every story in mid to high rise building structures. Hence, the idea of performing this scenario is to explore which DL model can yield the best performance in a relatively uncomplicated approach. Afterwards, the best performing model will be used to test its performance for a more challenging scenario, which is to assess damage in a structural system using only one single channel measurement installed on its first floor.

4.4.1 Experimental Settings

A numerical study is proposed to test the actual method applied to the 10 DOF building system. The damage level is selected to be 50%, which corresponds to a loss of stiffness of the main structural elements (column, beam, slab) of the corresponding story where the damage is simulated. All the proposed case studies aim to track the damage on a global floor wise, without detecting the exact location of damage, whether it is in a beam, shear wall, slab, or column. The damage level is selected to be 50% after considering other damage levels of 10% and 20%. The adopted damage level yielded the best performance. The damage simulated in the proposed case studies is a single damaged floor at a time; double or triple damaged stories are out of the scope of this study.

The building is simulated as a linear, classically damped, and discrete lumped-mass n degrees of freedom (DOFs) structural system, subjected to a wide-band random shake excitation input force, $u(t)$. **Equation 4.7** governs the structural response of the adopted system.

$$M\ddot{x}(t) + C\dot{x}(t) + Kx(t) = u(t) \quad (4.7)$$

Where $x(t)$ is a vector of displacement response at DOFs, M is the mass matrix, C is the damping matrix, and K is the stiffness matrix. The solution of **Equation 4.7** for any dynamical system can be formulated using the state-space model, as presented in equations **Equation 4.8**, **Equation 4.9**, and **Equation 4.10** shown below.

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.8)$$

$$\dot{\hat{x}} = A\hat{x} + Bu \quad (4.9)$$

$$y = \hat{C}\hat{x} + Du \quad (4.10)$$

Where A represents the state matrix, B is the input matrix, \hat{C} is the output matrix, and D is the transmission matrix. For the case of a 3-DOF, M , K , and C (Rayleigh Equation) are expressed in **Equation 4.11**, **Equation 4.12**, and **Equation 4.13**, respectively. **Figure 4.7** shows a three-dimensional representation of a three DOF building.

$$M = \begin{vmatrix} m1 & 0 & 0 \\ 0 & m2 & 0 \\ 0 & 0 & m3 \end{vmatrix} \quad (4.11)$$

$$K = \begin{vmatrix} K1 + k2 & -K2 & 0 \\ -k2 & K2 + K3 & -K3 \\ 0 & -K3 & K3 \end{vmatrix} \quad (4.12)$$

$$C = \alpha M + \beta K, \text{ where } \alpha \text{ and } \beta \text{ are the Rayleigh damping coefficients.} \quad (4.13)$$

Table 4.3 summarizes the adopted parameters for the calculation of the mass and stiffness matrix. The acceleration signals were collected under a random shake excitation at a sampling frequency of 1000 Hertz. Signals were captured for 256 s, so each signal comprises $n_T=256000$ samples. The excitation was simulated by a vector of size (256000 x 1) of normally distributed random numbers using the MATLAB function *randn* (256000,1). A data processing program was written in a DL platform, Pytorch (<https://pytorch.org>), to load the data from CSV files and arrange it into the appropriate form. Two forms of data classification were adopted; the first is a simple binary classification (Healthy/Damaged) without determining the exact location of damage, the second classifies the building as healthy or damaged and determines the precise location of the damage.

4.4.2 Evaluation Metrics

The comparison between the machine and DL models was based on various performance metrics, such as accuracy (**Equation 4.14**), precision (**Equation 4.15**), recall (**Equation 4.16**), *f1* score (**Equation 4.17**), Area Under the Receiver Operating Characteristic curve

(ROC_AUC) score, which represents the area under the roc curve ($TPR = f(FPR)$), sensitivity or True positive rate (**Equation 4.18**), False positive rate (**Equation 4.19**) and Matthew Correlation Coefficient (MCC) (**Equation 4.20**). The calculation of all these metrics was based on the Confusion matrix or Error Matrix to evaluate classification results since the precision of classification for the prediction task is not reliable if the set of inputs contains more than two classes (Kohavi *et al.*, 1998).

Table 4.3: Structural Parameters for the Three Numerical Models.

| Story | Mass (Kg) | Stiffness (N/m) | Damping Coefficient |
|------------------|-----------|-----------------|---------------------|
| 1 st | 1 | 1750 | 2% |
| 2 nd | 1 | 1575 | |
| 3 rd | 1 | 1400 | |
| 4 th | 1 | 1225 | |
| 5 th | 1 | 1050 | |
| 6 th | 1 | 875 | |
| 7 th | 1 | 700 | |
| 8 th | 1 | 525 | |
| 9 th | 1 | 350 | |
| 10 th | 1 | 175 | |

The confusion matrix, as illustrated in **Figure 4.8**, is a table layout used to visualize the performance of an algorithm. Each line of this table represents the instances of a predicted class, while each column represents the actual class.

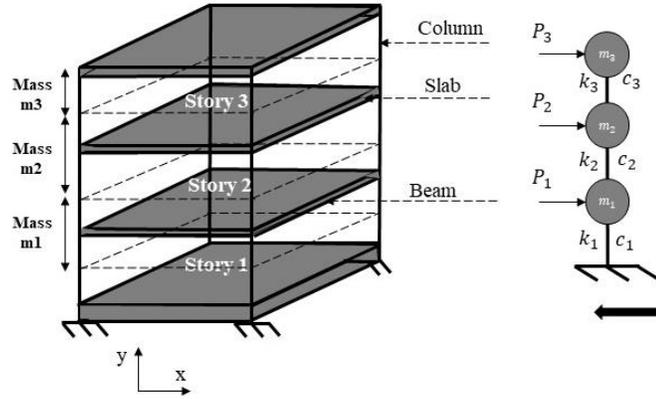


Figure 4.7: Three-Dimensional Model of a Three DOF System.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.14)$$

$$\text{Precision} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.15)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4.16)$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.17)$$

$$\text{Sensitivity} = \text{True positive rate} = \frac{TP}{TP + FN} \quad (4.18)$$

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \quad (4.19)$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \quad (4.20)$$

The reason for employing these metrics is that standard indicators such as "accuracy" and "precision" can be misleading when the data are imbalanced (Tharwat, 2018) since they use values from both columns of the confusion matrix. Thus, as data distribution changes, these metrics change as well, even if the classifier performance does not. Consequently, these two metrics are not sufficient to compare the proposed ML and DL models. It was reported that MCC outputs a more informative and truthful score than accuracy and *f1* score for classification models evaluation, first by explaining their mathematical features and

then by employing MCC in six synthetic use cases and real genomics scenario (Chicco and Jurman, 2020). This coefficient considers true and false positives and negatives, and it returns a value between -1 and 1. A ratio of 1 represents a perfect prediction, 0 random predictions, and -1 a total disagreement between prediction and observation.

| | | Actual Class | |
|-----------------|---|--------------|----|
| | | P | N |
| Predicted Class | P | TP | FP |
| | N | FN | TN |

Figure 4.8: Confusion Matrix Terminology.

4.4.3 Case Study 1

4.4.3.1 Definition

To determine the best model applicable to the first damage assessment scenario, a comparative study between the most prominent ML and DL models for time series classification is proposed. The first case study is considered the least complicated in terms of the damage assessment procedure, which consists of installing a network of sensors on each floor to track the damage separately. This case study implements ten sensors on ten floors with eleven classes (one healthy configuration and ten damaged classes corresponding to the tracked damage on each floor, separately).

4.4.3.2 Comparative Study

To assess the complexity of the proposed scenario and its corresponding dataset, a comparison between ML and DL models is provided. Four conventional ML models, including i.) Decision Trees (DTs) which are a non-parametrical supervised method of learning used to regress and classify. DTs learn from data to approximate the sinus curve with a set of conditions. The broader the tree, the more complex the rules of decision and

the more fit the model (Quinlan, 1986); ii) Bootstrap aggregating (Bagging), which is an ML ensemble meta-algorithm designed to boost the stability and accuracy of ML algorithms for classification and regression tasks. It is used to reduce the variance and to avoid over-fitting. It is applied to the DT model (Breiman, 1996); iii) K Nearest Neighbors (KNN), which are a basic ML algorithm that stores all existing cases and categories of new cases based on a measure of similarity (distance). KNN is a nonparametric technique for classification and regression tasks. The greater K , the more precise is the classification, but the longer it takes for the task to be performed (Cunningham *et al.* 2020); iv) Random Forest (RF), which is a classification algorithm with multiple decision trees. It incorporates Bagging and randomness when assembling each tree to establish an uncorrelated forest of trees whose estimation is more reliable than any single tree (Breiman *et al.*, 2001).

For the ML part, three main steps are required; the first step is to preprocess the signals by denoising them. However, no filtering operation is necessary since the data is obtained from a numerical model, and it is extracted from a controlled environment. Secondly, manual extraction of the central damage sensitive features from the raw acceleration signals is proposed. These extracted damage sensitive features include the root mean square of the raw acceleration signals, the variance, the maximum amplitude, the skew, the kurtosis, the wave, the spectral kurtosis, the spectral skew, and the spectral power of the raw acceleration signals.

Finally, these features are fed into the proposed ML models. On the other hand, for DL models, the proposed model, BuildingNet_1, is compared to two DL algorithms, which are: i) Long Short-Term Memory (LSTM), which are modified variants of RNNs, making it easier to recall memory details. They are used to solve RNN's vanishing gradient problems.

LSTM proved useful in classifying, processing, and predicting time-series given time lags of unknown duration. They are trained using backpropagation (BP), which helps to optimize the outputs of the layers.

RNN is a feedforward neural network that has internal memory. RNN is recursive as it executes the same operation for each data input, while the new input-output relies on the

previous computation. Upon processing the output, it is copied back to the recurrent network; ii) Residual Networks, which extends the neural networks to profound structures by adding the shortcut connection in each residual block to enable the gradient flow directly through the bottom layers (Fawaz *et al.*, 2019). The difference with the standard convolutions is that a direct shortcut is added to link the output of a residual block to its input. Thus, this should enable the flow of the gradient directly through these connections, which makes training a deep neural network much more accessible by reducing the vanishing gradient effect (Wang *et al.*, 2017). This architecture is considered the most profound architecture with the first nine of its 11 layers being convolutional, followed by a global pooling layer that averages the time series across the time dimension. This model can be trained on a specific dataset, then transfer and fine-tune it on a target dataset without the need for modifying the hidden layers of the architecture.

4.4.3.3 BuildingNet_1 Prediction Performance

Table 4.4 presents a comparison of the testing results of the proposed ML and DL models for the first assessment damage scenario, which consists of installing ten sensors, of monitoring the damage separately on each floor. The comparison between the ML models based on a combination of accuracy, *f1* score, and MCC proves that RF was the most suitable classifier for the proposed dataset among all the other ML models. However, ML models were still unable to learn the proposed dataset. For this reason, the DL part has been discussed in terms of three main models, as presented above.

Table 4.4: Comparison between conventional ML and DL Time-Series classifiers using different metrics corresponding to the first damage assessment scenario.

| Metrics | ML models | | | | DL models | | |
|----------|-----------|---------|-------|-------|-----------|------|-------------|
| | DT | Bagging | KNN | RF | ResNet | LSTM | 1D-CNN |
| Accuracy | 0.463 | 0.509 | 0.494 | 0.516 | 0.69 | 0.74 | 0.89 |
| F1 score | 0.594 | 0.633 | 0.618 | 0.638 | 0.77 | 0.80 | 0.9 |
| ROC_AUC | | | | | 0.905 | 0.95 | 0.96 |
| MCC | 0.084 | 0.106 | 0.112 | 0.115 | 0.22 | 0.33 | 0.40 |

It can be observed that the 1D CNN model developed in this study as "BuildingNet_1" was the best for all the ML metrics. For instance, the overall accuracy of the model was about 90%, compared to 74% and 69% for ResNet and LSTM, respectively. The same trend was observed for the other metrics, including the weighted average *f1* score, MCC, and especially the ROC_AUC score, which was 96%, confirming that it is an excellent model. "BuildingNet_1" is an excellent fit for the first damage assessment scenario.

Table 4.5 shows the confusion matrix of "BuildingNet_1", which presents 11 classes; the first class corresponds to the healthy class, while D1 till D10 are the successive damaged stories from the beginning to the last story. As explained in Section 3, the healthy class is much more significant than the other classes. For instance, the support for the actual healthy class is 2500 compared to almost 25 per damaged class. Thus, some misclassifications can be related to the healthy class, and this can be explained by the fact that the damaged classes (less support) were biased by the healthy class (more support). However, this does not affect the overall performance of the classifier, as indicated by the ML metrics.

Table 4.5: BuildingNet_1 Confusion Matrix.

| | Actual Label | | | | | | | | | | | |
|---------|--------------|----|----|----|----|----|----|----|----|----|-----|--|
| | Healthy | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | |
| Healthy | 2296 | 6 | 7 | 9 | 16 | 10 | 8 | 7 | 6 | 10 | 3 | |
| D1 | 15 | 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| D2 | 32 | 0 | 16 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| D3 | 6 | 0 | 0 | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| D4 | 9 | 1 | 1 | 0 | 16 | 1 | 0 | 0 | 0 | 0 | 0 | |
| D5 | 12 | 0 | 0 | 0 | 0 | 23 | 0 | 1 | 0 | 0 | 0 | |
| D6 | 17 | 0 | 0 | 0 | 0 | 0 | 13 | 0 | 1 | 0 | 0 | |
| D7 | 24 | 0 | 0 | 0 | 0 | 0 | 1 | 17 | 1 | 0 | 0 | |
| D8 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | |
| D9 | 37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 10 | 0 | |
| D10 | 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | |

4.4.4 Case Study 2

4.4.4.1 Definition

Unlike case study 1, which determines the appropriate DL model for assessing damage in a less complicated scenario consisting of installing a sensor on each floor, this case study 2 aims at tracking damage in each floor of the building system using only one sensor. The same DL model was used to determine damage in a more complicated situation, but with a different configuration of the network's parameters and hyperparameters, as illustrated in **Tables 4.1** and **4.2**. Contrary to case 1, all classes were nearly equal, and no data imbalance was encountered in this case. Thus, accuracy and *f1* score should be enough to assess the efficiency of the proposed model.

4.4.4.2 BuildingNet_2 Prediction Performance

It can be observed from **Figure 4.9** and **Tables 4.6** and **4.8** that the model can yield a high *f1* score and accuracy exceeding 91% and 98%, respectively, in assessing damage on each floor separately. However, starting from the seventh floor, the model's performance started to drop. For instance, moving from the 6th to the 7th floor, the *f1* score decreased from 91% to 84% and finally reached 34% to assess the damage on the final story (10th floor).

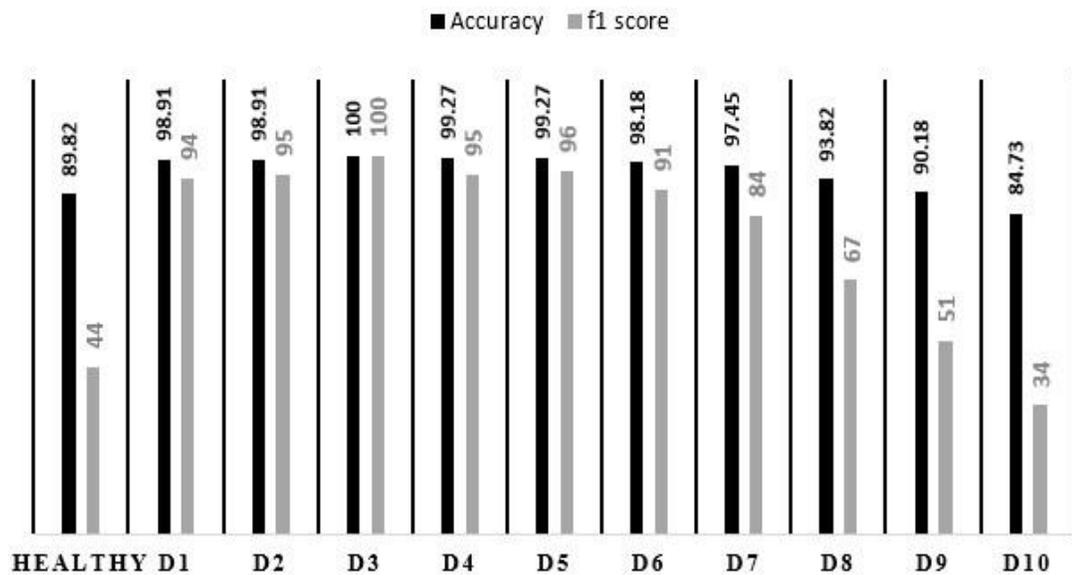


Figure 4.9: Evaluation of BuildingNet_2 Damage Distribution in Terms of Accuracy and f1 score.

Moreover, the healthy class had some misclassifications due to the above-noted decrease in performance between floors 7 and 10, leading the model to misclassify the damaged 10th floor as a healthy one. This result is rather expected because of the long distance between the first floor where the sensor is installed and the tenth, ninth or eighth floors where damage was located. The overall performance of the proposed 1D CNN model is reasonable for the scenario of having only one sensor to separately assess the damage on each floor and precisely localize it. This can serve as a baseline model for localizing damage in Multi-DOF systems and can yield considerable performance in localizing damage on the first six floors. However, the inspector should pay extra attention to

detecting damage on the upper floors since results could be misleading. A proposed work installed several sensors on selected floors, called measurement floors (Rafiei and Adeli, 2017). These floors were defined as locations of stiffness or mass changes. They divided a scaled 38 floor reinforced concrete high-rise building into nine measurement floors in a way that for every five stories, they need an additional sensor to track damage. However, their approach consisted just of assessing the overall health state of the structure without determining the exact position of the floor where the damage was located.

Table 4.6: BuildingNet_2 Confusion Matrix.

| | Actual Label | | | | | | | | | | | |
|---------|--------------|----|----|----|----|----|----|----|----|----|-----|--|
| | Healthy | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | |
| Healthy | 11 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 2 | 8 | |
| D1 | 0 | 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| D2 | 0 | 1 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| D3 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| D4 | 0 | 0 | 0 | 0 | 18 | 1 | 1 | 0 | 0 | 0 | 0 | |
| D5 | 0 | 0 | 0 | 0 | 0 | 22 | 0 | 0 | 1 | 0 | 0 | |
| D6 | 0 | 0 | 0 | 0 | 0 | 0 | 26 | 0 | 0 | 0 | 3 | |
| D7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 1 | 0 | 2 | |
| D8 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 17 | 3 | 2 | |
| D9 | 3 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 14 | 5 | |
| D10 | 8 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 10 | 11 | |

4.4.5 Case Study 3

4.4.5.1 Definition

In the second case study, "BuildingNet_2" demonstrated an accuracy of 75% in assessing the position of damage in the ten DOF system. To illustrate the excellent performance of

the proposed model, a simpler damage assessment scenario that aims to classify the overall state of the building as damaged or not, was explored.

4.4.5.2 BuildingNet_3 Prediction Performance

As illustrated in **Tables 4.7** and **4.8**, the proposed DL model "BuildingNet_3" proved its effectiveness in detecting whether the structural system is damaged or not, reaching an accuracy of about 95%. Thus, the proposed network can be used in a preliminary analysis to determine the global health state of the structural system accurately and effectively. As a result, installing one sensor to track the overall health condition, had perfectly proved its high applicability when dealing with this range of building height, and it is expected to serve as a preliminary assessment to check whether a building is damaged or not.

Table 4.7: BuildingNet_3 confusion matrix.

| | | Actual Label | |
|-----------------|---------|--------------|---------|
| | | Healthy | Damaged |
| Predicted Label | Healthy | 19 | 8 |
| | Damaged | 6 | 242 |

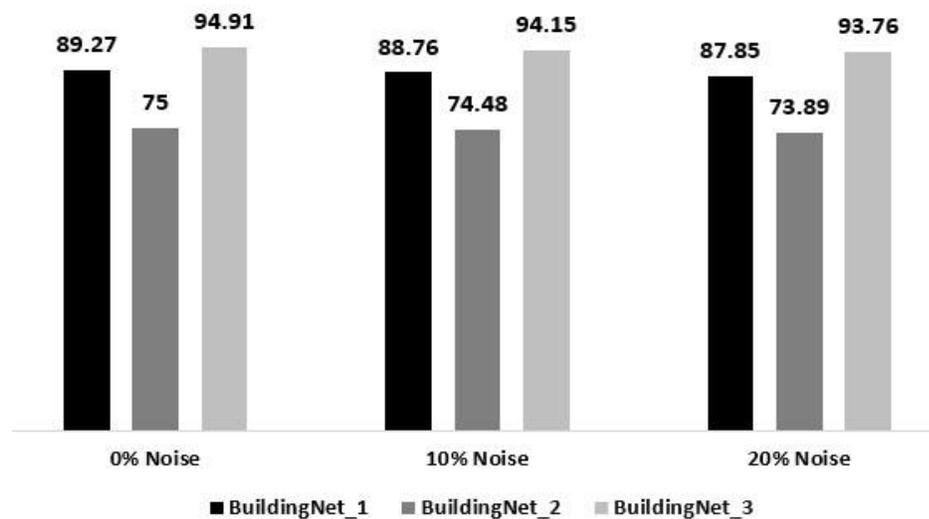


Figure 4.10: Comparison of Accuracy Results with 0-20% Random Gaussian Noise Added to the Testing Data.

4.4.6 Effect of Noise

To test the robustness of the proposed DL models, their sensitivity up to 20% noise was investigated. Both original and noisy data were arranged in the comparative tests. The definition of noise level here is the ratio of white Gaussian noise injected in all the three datasets corresponding to the three proposed models, in a way that the networks were trained and tested using noisy data. **Figure 4.10** shows the accuracies of the three scenarios. It can be observed that "BuildingNet_1" achieved high efficiency of 94.15% in the ten percent noise situation.

Furthermore, it scored superior performance even for the 20% noise case with an accuracy of 93.76%. It can be deduced that when white noise was injected into the data set, the proposed models still achieved considerable accuracy with a margin of difference, not exceeding 2% for all the proposed models. The results show that the damage assessment using *BuildingNet* models was reliable even when the test data were smeared up to a 20% noise ratio, which confirms the robustness of the proposed models.

However, it is worth mentioning that the second case study needs more consideration in the future, despite its practical use consisting of installing one sensor to track the damage on each damaged floor separately, which serves to limit the number of sensors installed into a structure to assess its health condition. For instance, installing one sensor to track the damage separately can be applied successfully and with high performance for the first six floors. This can serve to limit the number of sensors to one sensor every six floors, and these floors are called measurement floors. Thus, it is suggested to follow this assumption when assessing damage using only one single channel measurement for different heights of the building. If the height of the building is less than six floors, using one sensor is expected to yield excellent performance. However, when exceeding six floors, one additional sensor per each additional six floors is needed to separately detect the damage on each floor accurately.

Table 4.8: Demonstration of Obtained Results for the Three Main DL Models Adopted in this Study.

| Metrics | Proposed Models | | |
|---------------------------|-----------------|--------------|--------------|
| | Case Study 1 | Case Study 2 | Case Study 3 |
| Overall Accuracy | 0.890 | 0.750 | 0.949 |
| Weighted Average f1 score | 0.900 | 0.760 | 0.731 |
| MCC | 0.400 | 0.736 | 0.703 |

4.5 Conclusion and Future Work

In this study, a real-time DL vibration-based technique for assessing structural damage in a 10 DOF system is proposed. Three damage assessment scenarios were illustrated using three different 1D CNN architectures. The first damage assessment scenario tracks the exact location of damage using ten sensors installed on each floor, and the second is similar to the first except using only one sensor mounted on the first floor. The latter only evaluates the condition of the structural system via a binary classification, healthy or damaged, without determining the exact location of the damage. The time record adopted for each experiment was 256s, with a sampling frequency of 1000 Hz. The selected training to testing ratio was 90:10. The SMOTE oversampling technique was used to augment the data to assist the DL models in the training phase, which helped to improve the performance of the testing phase. Based on the experimental findings, the following conclusions can be drawn:

- The proposed 1D CNN model can locate structural damage separately by installing one sensor on each floor, with an accuracy of about 90%.
- The baseline model for detecting damage on each floor using one single channel measurement, proposed in this study reached an accuracy of 75%.
- The binary classification of damage proposed in this study demonstrated superior performance and accuracy of about 95%.

- The performance of the 1D CNN classifier in such a scenario was superior for the first six to seven stories, reaching more than 97% accuracy and 91% for the *f1* score. However, starting from the 7th floor, the model began to misclassify some damage patterns, reaching about 85% and 34% for accuracy and *f1* score, respectively.
- It is believed that installing one sensor at every six stories (substructures or measurement floors) could be a functional solution to mitigate the limitations of the proposed study in high-rise buildings, which can be more cost-effective and practical.
- The proposed models demonstrated excellent performance when adding up to 20% random Gaussian noise ratio, with no more than 2% decrease in models' accuracy.

4.6 References

- Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., & Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, 388, 154-170.
- Abdeljaber, O., & Avci, O. (2016). Nonparametric structural damage detection algorithm for ambient vibration response: Utilizing artificial neural networks and self-organizing maps. *Journal of Architectural Engineering*, 22(2), 04016004.
- Abellan-Nebot, J. V., & Subirón, F. R. (2010). A review of machining monitoring systems based on artificial intelligence process models. *The International Journal of Advanced Manufacturing Technology*, 47(1-4), 237-257.
- Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.
- Bao, Y., Tang, Z., Li, H., & Zhang, Y. (2019). Computer vision and deep learning-based data anomaly detection method for structural health monitoring. *Structural Health Monitoring*, 18(2), 401-421.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
- Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2), 123-140.
- 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.

- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 6.
- Cunningham, P., & Delany, S. J. (2020). k-Nearest Neighbour Classifiers---. *arXiv preprint arXiv:2004.04523*.
- Doebbling, S. W., Farrar, C. R., Prime, M. B., & Shevitz, D. W. (1996). Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review (No. LA-13070-MS). Los Alamos National Lab., NM (United States).
- Farrar, C. R., & Worden, K. (2007). An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *365*(1851), 303-315.
- Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, *33*(4), 917-963.
- Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, *61*, 863-905.
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
- Iqbal, W., Berral, J. L., & Carrera, D. (2020). Adaptive sliding windows for improved estimation of data center resource utilization. *Future Generation Computer Systems*, *104*, 212-224.
- Khan, A., Ko, D. K., Lim, S. C., & Kim, H. S. (2019). Structural vibration-based classification and prediction of delamination in smart composite laminates using deep learning neural network. *Composites Part B: Engineering*, *161*, 586-594.
- Khodabandehlou, H., Pekcan, G., & Fadali, M. S. (2019). Vibration-based structural condition assessment using convolution neural networks. *Structural Control and Health Monitoring*, *26*(2), e2308.

- Kim, B., & Cho, S. (2019). Image-based concrete crack assessment using mask and region-based convolutional neural network. *Structural Control and Health Monitoring*, 26(8), e2381.
- Kiranyaz, S., Ince, T., & Gabbouj, M. (2015). Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, 63(3), 664-675.
- Kiranyaz, S., Ince, T., Abdeljaber, O., Avci, O., & Gabbouj, M. (2019, May). 1-d convolutional neural networks for signal processing applications. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 8360-8364). IEEE.
- Kohavi, R., & Provost, F. (1998). Glossary of terms journal of machine learning. *Mach. Learn.*
- Oh, B. K., Lee, S. H., & Park, H. S. (2020). Damage localization method for building structures based on the interrelation of dynamic displacement measurements using convolutional neural network. *Structural Control and Health Monitoring*, e2578.
- Pathirage, C. S. N., Li, J., Li, L., Hao, H., Liu, W., & Ni, P. (2018). Structural damage identification based on autoencoder neural networks and deep learning. *Engineering Structures*, 172, 13-28.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1), 81-106.
- Rafiei, M. H., & Adeli, H. (2017). A novel machine learning-based algorithm to detect damage in high-rise building structures. *The Structural Design of Tall and Special Buildings*, 26(18), e1400.
- Suykens, J. A., & Vandewalle, J. (1999). Chaos control using least-squares support vector machines. *International journal of circuit theory and applications*, 27(6), 605-615.
- Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*.
- Wang, Z., Yan, W., & Oates, T. (2017, May). Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)* (pp. 1578-1585). IEEE.

- Xu, Y., Wei, S., Bao, Y., & Li, H. (2019). Automatic seismic damage identification of reinforced concrete columns from images by a region-based deep convolutional neural network. *Structural Control and Health Monitoring*, 26(3), e2313.
- Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 14(1), 35-62.
- Zhang, T., Biswal, S., & Wang, Y. (2019). SHMnet: condition assessment of bolted connection with beyond human-level performance. *Structural Health Monitoring*, 1475921719881237.
- Zhang, Y., Sun, X., Loh, K. J., Su, W., Xue, Z., & Zhao, X. (2020). Autonomous bolt loosening detection using deep learning. *Structural Health Monitoring*, 19(1), 105-122.

Chapter 5

5 Conclusions, Recommendations, and Future Research

5.1 Conclusions

There has been a rapid increase in the volume of research on applications of machine learning algorithms in the field of structural health monitoring. The application of ML algorithms to detect, assess, and possibly repair and rehabilitate damage in civil engineering structures is garnering increasing attention.

Chapter two presented, critically surveyed, discussed, and analyzed the main techniques and algorithms that have been deployed for this purpose in the open literature. Detailed tables have been created to summarize the state-of-art and provide the reader with convenient access to the volume of work that has been conducted in this domain. The advantages and limitations of these techniques have been identified, and best practice recommendations for their use have been formulated. Knowledge gaps and the needed future research have been outlined.

Chapter three proposed an automated inspection model for concrete structures using DL and IPTs to detect cracks. A convolutional neural network was trained independently on an image database consisting of 40k images with a 227x227 pixel resolution. The used classifier englobed five classes based on two criteria: the condition of the concrete surface (presence of cracks), and the orientation of cracks (HR, HL, VR, and VL). The total number of images used for training and testing of the classifier was 6000, with a split of 60:20:20 (3600 Images for training and 1200 images for validation and testing). IPTs have been implemented to induce transformations in the pictures tested by the CNN classifier for localizing cracks using Otsu's method for thresholding, noise removal, image binarization, and image segmentation. After localizing the crack, its geometrical properties, including length, width, and angle of orientation, are calculated. A field test was conducted to evaluate the performance of the trained IPT algorithm by testing images of higher pixel resolution.

In Chapter four, a real-time DL vibration-based technique for assessing structural damage in a 10 DOF system was proposed. Three damage assessment scenarios were illustrated using three different 1D CNN architectures. The first damage assessment scenario tracks the exact location of damage using ten sensors installed on each floor, and the second is similar to the first except using only one sensor mounted on the first floor. The latter only evaluates the condition of the structural system via a binary classification, healthy or damaged, without determining the exact location of the damage. The time record adopted for each experiment was 256s, with a sampling frequency of 1000 Hz. The selected training to testing ratio was 90:10. The SMOTE oversampling technique was used to augment the data to assist the DL models in the training phase, which helped improving the performance of the testing phase. The proposed 1D CNN model can locate structural damage separately by installing one sensor on each floor, with an accuracy of about 90%. The baseline model for detecting damage on each floor using one single channel measurement, proposed in this study, reached an accuracy of 75%. The binary classification of damage proposed in this study demonstrated superior performance and accuracy of about 95%.

This study offers a nearly fully automated modern inspection platform that combines the most recent engineering interdisciplinary subfields, including the application of Machine Learning, Deep Learning, Image Processing, and Signal processing techniques. The obtained results prove that this study is time and cost-saving.

5.2 Recommendations and Future Research

While the current applications of deep learning in SHM are mainly limited to global damage classification tasks, there is minimal research on the quantification and exact localization of damage in terms of crack length, width, and degree of severity in the open literature. Moreover, unlike other studies, the work conducted in this thesis proposes a decentralized approach that aims to detect damage in building structures using a single based channel measurement, which is expected to limit the costs of sensors installation drastically. Thus, the work reported in this thesis blazes the trail for future research and further studies as follows:

1. For the vibration-based model, a field test needs to be conducted to assess the performance of the proposed models in real-life structures subjected to different kinds of loadings, which can establish the proposed approach as a reliable inspection platform for the vibration-based aging portfolio of building structures
2. For the vision-based model, it is believed that with further developmental work, and coupling it with UAV acquired images, the developed method can offer an automated platform for inspection of the colossal backlog of aging concrete structures worldwide
3. More concrete durability issues in terms of visual defects should be covered (e.g., efflorescence, spalling, scaling), and this can be done by providing more datasets for images with specific damage patterns.
4. Damage quantification in terms of volume spalling should be investigated, which can give better insight into the hidden features of damage inside concrete structural elements.
5. Visual damage quantification can be coupled with X-ray scanning, infrared thermography, and/or ground penetrating radar to give more information on the physical degradation mechanisms in concrete.
6. More construction and building materials should be investigated in terms of visual damage (Steel, wood, masonry) and more civil engineering structures should take part in the vibration-based study by expanding the research to other facilities (bridges, dams, tunnels, pipelines, etc.)
7. One promising research direction would be combining the vision-based model and the vibration-based model in only one process. This is expected to offer a fully reliable inspection platform with minimal intervention from the human inspectors. With further validation work on real engineering structures, this study could be marketed to the engineering industry to transform civil engineering inspection protocols.

Curriculum Vitae

Name: Majdi Flah

Post-secondary Education and Degrees: National Engineering School of Gabes
Gabes, Tunisia
2013-2018 B.A.

The University of Western Ontario
London, Ontario, Canada
2018-2020 M.A.

Honours and Awards: Teaching Assistant Excellence Award
2019-2020

Mitacs Graduate Fellowship
Master Fellowship
2018-2019

Mitacs Globalink Scholarship
Bachelor Scholarship
2017-2018

Related Work Experience Teaching Assistant
The University of Western Ontario
2019-2020

Publications:

Flah, M., Itzel, N., Chaabene, WB., and Nehdi, M. L. “Machine Learning Algorithms in Civil Structural Health Monitoring: A Systematic Review,” submitted to *Archives of Computational Methods in Engineering*, 2020.

Flah, M., Suleiman, AR., and Nehdi, M. L. “Classification and Quantification of Cracks in Concrete Structures using Deep Learning Image-Based Techniques,” submitted to *Cement and Concrete Composites*, 2020.

Flah, M., and Nehdi, M. L. (2020) “Automated Crack Identification using Deep Learning Based Image Processing,” National Conference on Infrastructure Management, Proceedings of the Annual Conference of the Canadian Society for Civil Engineering, Saskatoon, Canada.

Flah, M., Ragab, M., Lazhari, M., and Nehdi, M. L. “Localization and Classification of Structural Damage using Deep Learning Single-Channel Signal-Based Measurement,” submitted to *Structural Health Monitoring*, 2020.

Chaabene, WB., **Flah, M.**, and Nehdi, M. L. “Machine Learning prediction of mechanical properties of concrete: Critical review,” *Construction and Building Materials*, Vol. 260, pp. 1-18, 2020.

Itzel, N., Marani, A., **Flah, M.**, and Nehdi, M. L. “Machine Learning Prediction of Compressive Strength of Modern Concrete: Systematic review”, submitted to *Cement*, Elsevier, 2020.