Advanced imaging techniques for damage characterization of concrete

A Thesis Submitted to
Nirma University
In Partial Fulfilment of the Requirements for
The Degree of
Doctor of Philosophy
in
Technology & Engineering

By
KAPADIA HARSH KHODIDAS
(16PTVPHDE164)

Electronics and Instrumentation Engineering Department
Institute of Technology, Nirma University
Ahmedabad, Gujarat, India
November 2022

Nirma University

Institute of Technology

Certificate

This is to certify that the thesis entitled **Advanced imaging techniques for damage characterization of concrete** has been prepared by **Mr. Kapadia Harsh Khodidas (16PTVPHDE164)** under my supervision and guidance. The thesis is his own original work completed after careful research and investigation. The work of the thesis is of the standard expected of a candidate for Ph.D. Programme in **Instrumentation and Control Engineering** and I recommend that it be sent for evaluation.

Date: 17|11|2022

Dr. Jignesh B. Patel,
Guide

P.v.Patel.

Dr. Paresh V. Patel
Co-Guide

Forwarded through:

(i) Head, Electronics and Instrumentation Engineering Department

(ii) Dean, Faculty of Technology and Engineering

19-11-'22

(iii) Dean, Faculty of Doctoral Studies and Research

To:

Executive Registrar
Nirma University

19/11/22

Nirma University
Institute of Technology

Declaration

I, **Kapadia Harsh Khodidas**, registered as Research Scholar, bearing Registration No. **16PTVPHDE164** for Doctoral Programme under the **Faculty of Technology and Engineering** of Nirma University do hereby declare that I have completed the course work, pre-synopsis seminar and my research work as prescribed under R. Ph.D. 3.5.

I do hereby declare that the thesis submitted is original and is the outcome of the independent investigations / research carried out by me and contains no plagiarism. The research is leading to the discovery of new facts / techniques / correlation of scientific facts already known. This work has not been submitted to any other University or Body in quest of a degree, diploma or any other kind of academic award.

I do hereby further declare that the text, diagrams or any other material taken from other sources (including but not limited to books, journals and web) have been acknowledged, referred and cited to the best of my knowledge and understanding.

Date: 17/11/2022

Kapadia Harsh Khodidas
(16PTVPHDE164)

I endorse the above declaration made by the student.

Date: 17/11/2022

Dr. Jignesh B Patel
Guide

P.V. Patel.

Dr. Paresh V Patel
Co-Guide

Enclosure - V

Specimen 'D'

Nirma University
Institute of Technology

Certificate

This is to certify that the thesis entitled **Advanced imaging techniques for damage characterization of concrete** has been prepared by me, under the supervision and guidance of **Dr Jignesh B Patel and Dr Paresh V Patel**. The thesis is my own original work completed after careful research and investigation. The work of the thesis is of the standard expected of a candidate for Ph.D. Programme. The final hard bound copy of the thesis is submitted after incorporating all the suggestions / corrections suggested by the referees.

Date: 11/08/2023

Kapadia Harsh Khodidas
(16PTVPHDE164)

Date: 11|8|2023

Dr. Jignesh B Patel, (Guide)

P.v.Patel
Dr. Paresh V Patel, (Co-Guide)

Forwarded Through:

(i) Dr. R N Patel,
Dean - Faculty of Technology and Engineering

(ii) Prof. (Dr.) P N Tekwani,
Dean - Faculty of Doctoral Studies and Research

(iii) Shri G R Nair,
Executive Registrar, Nirma University

Acknowledgement

I am thankful to the almighty for my existence, mother nature for the environment I live in and our great country India where I exist today.

I would like to express my deepest gratitude to Dr Jignesh B Patel and Dr Paresh V Patel for guiding me through the journey of my PhD. I would also like to thank them for giving me the opportunity to work under their supervision. I will be forever indebted to them for their constant inspiration, supervision, and support. All the discussions with them during the work have provided the best knowledge, wisdom and thinking ability. The journey has immensely enhanced my overall intellectual ability, improved my thought process and provided clarity in whatever I do.

I would like to express my sincere gratitude towards the RAC members Dr Manjunath Joshi, the late Dr Tanish Zaveri and Dr Priyank Thakkar for their valuable comments, suggestions and guidance. I sincerely thank the Head of the Electronics and Instrumentation Engineering Department Dr Himanshu K Patel and Dr Dipak M Adhyaru for their cooperative support and motivation. I would also thank all the colleagues and non-teaching staff members of the department for their cooperation and for maintaining a supportive environment. I would also like to sincerely thank Director, ITNU, Director, DRI-NU, the management and higher authorities of Nirma University for providing the opportunity to pursue a PhD. In addition to that, I would also like to thank the members of PhD section of the university for their smooth support in the entire phase of my PhD.

I would also like to thank my teachers Mr Dharmesh Naik, Mr C P Rathi, Mr Maurya Shah, Dr Vipul Shah, Dr Jayesh Barve, and Dr Bhanuprasad Pinnamaneni whom I have idolised in the life and imbibed their knowledge and learnings within me.

I would like to take this opportunity to thank Prof Alpesh Patel, who has been a constant source of motivation and a critic during the journey. Prof Alpesh, Prof Digesh and Prof Sunil have massively supported the work carried out during the PhD journey. I would like to sincerely thank Mr Sunil Regar and Mr Prahlad Raval for supporting the concrete cube testing in the laboratory. Finally, I would like to thank my students Maulik Vala, Ripalkumar Patel, Yash Shah, Shreyash Suratwala, Rushika Jani, Yash Richhariya, Shivam Patidar, Narotam Sharma, Shubham Vithlani, Haard Mehta, Pranav Agrawal, Pushpit Jain, Rishi Hiran, Nimit Soneji, Anirudha Rotti, and others who have assisted me during the work.

I am extremely blessed to have received boundless love, care, inspiration and backing from my mother Ramilaben Makwana, father Khodidas Makwana, elder brother Raj Kapadia, my wife Alka Kapadia, sister-in-law Roshni Kapadia, nephew Shiv and Trisha and my daughter Shaivee throughout the journey. They have wholeheartedly supported me through the extensive working hours and had been there in every situation. I will be forever indebted to them. I also want to thank my friends, family members, and in-laws for their support and motivation.

Finally, I would like to express my sincere gratitude to one and all who have directly or indirectly supported, contributed, and motivated me in the work.

List of Tables

Table 1. Vision hardware specifications. 66

Table 2. Instrumentation hardware specifications. 69

Table 3. ROC analysis. 125

Table 4. Performance analysis of classifier using KNN. 134

Table 5. Performance evaluation of CNN tested on a binary dataset. 137

Table 6. Performance evaluation of CNN tested on imbalanced greyscale dataset. 142

Table 7. Performance evaluation of CNN tested on a balanced greyscale dataset. 144

Table 8 Hyperparameters of CNNs. 148

Table 9. Comparative analysis of crack detection performance of implemented methods. 148

Table 10. Performance comparison of different methods from the literature. 150

Table 11. Inference time analysis on CPU alone. 154

Table 12. Inference time analysis on CPU + TPU. 155

Table 13. Summary of compression tests on concrete cubes. 162

Table 14. Summary of crack monitoring and analysis. 164

Table 15. Crack load analysis of a concrete cube-A. 167

Table 16. Progression details of crack numbers 1 and 2 in concrete cube – A. 171

Table 17. Progression details of crack numbers 3 and 4 in concrete cube – A. 171

Table 18. Progression details of crack numbers 5 and 6 in concrete cube – A. 172

Table 19. Load versus crack monitoring data for concrete cube-A. 173

Table 20. Crack load analysis of a concrete cube-B. 174

Table 21. Progression details of crack numbers 1 and 2 in concrete cube – B. 178

Table 22. Progression details of crack numbers 3 and 4 in concrete cube – B. 178

Table 23. Progression details of crack numbers 5 and 6 in concrete cube – B. 179

Table 24. Progression details of crack number 7 in concrete cube – B. 179

Table 25. Load versus crack monitoring data for cube-B. 180

Table 26. Crack load analysis of a concrete cube – C. 181

Table 27. Progression details of crack numbers 1 and 2 in concrete cube – C. 186

Table 28. Progression details of crack numbers 3 and 4 in concrete cube – C. 186

Table 29. Progression details of crack numbers 5 and 6 in concrete cube – C. 187

Table 30. Progression details of crack numbers 7 and 8 in concrete cube – C. 187

Table 31. Progression details of crack number 9 in concrete cube – C. 188

Table 32. Load versus crack monitoring data for concrete cube-C. 188

Table 33. Crack load analysis of a concrete cube-D. 189

Table 34. Progression details of crack numbers 1 and 2 in concrete cube - D. 192

Table 35. Progression details of crack numbers 3 and 4 in concrete cube – D. 193

Table 36. Progression details of crack number 5 in concrete cube – D. 193

Table 37. Load versus crack monitoring data for concrete cube-D. 194

List of Figures

Figure 1. Sample concrete crack images..29

Figure 2 Samples of cracks in concrete with variations. 31

Figure 3. Crack width ruler (elcometer-143-crack-width-ruler , 2022)..................................38

Figure 4. Tools to measure crack width..38

Figure 5 Sample and results images of percolation method. (Tomoyuki Yamaguchi , 2008) 42

Figure 6 Sample and results images of line filter method. (Yusuke Fujita , 2006) 43

Figure 7 Crack detection results of HTF method. (Han Hu , 2010) 44

Figure 8 Result images of multiple filtering and threshold method. (Ahmed Mahgoub Talab , 2016) 45

Figure 9 Result images of peak detection method. (Tran Hiep Dinh , 2016)..........................46

Figure 10. Classification of vision-based methods to detect cracks in concrete. 48

Figure 11. Crack detection methodology block diagram...63

Figure 12. Vision hardware..68

Figure 13. Instrumentation hardware..70

Figure 14. Experimental setup..73

Figure 15. Concrete cube surface images without illumination and experimental setup. 74

Figure 16. Concrete cube surface images with illumination and experimental setup. 76

Figure 17. Sample images of a concrete cube captured during a compression test................78

Figure 18. Load versus time graph of a concrete cube. 79

Figure 19. Sample images of multiple concrete cubes captured during the compression test.81

Figure 20. Load versus time graph of multiple concrete cubes. 82

Figure 21. Crack detection methodology..85

Figure 22. Flow chart of image pre-processing and crack detection. 87

Figure 23 Flowchart of the crack detection method 1. 89

Figure 24. Algorithm of crack detection method 1...90

Figure 25. Flowchart of percolation process..92

Figure 26 Algorithm of percolation based crack detection...93

Figure 27 Flowchart of training KNN and SVM models. 96

Figure 28. Architecture of convolutional neural network...98

Figure 29. Flowchart of dataset generation..102

Figure 30 Sample images from SDNET dataset. (Structural Defects Network (SDNET) 2018) 103

Figure 31 Snapshot of CODEBRIM dataset (meta-learning-CODEBRIM , 2022)..............105
Figure 32. Flowchart of dataset generation using binary contour images. 107
Figure 33. Sample images of non-crack binary contours. 108
Figure 34. Sample images of crack binary contours. 108
Figure 35. Flowchart of dataset generation using greyscale contour images. 109
Figure 36. Sample images of non-crack greyscale contours...110
Figure 37. Sample images of crack greyscale contours. 110
Figure 38. Augmented images of a sample crack contour image. 111
Figure 39 Architecture of Inceptionv3. 112
Figure 40. Flowchart of CNN model training using binary image dataset. 113
Figure 41. Flowchart of CNN model testing using binary image dataset. 114
Figure 42. Flowchart of CNN model training using greyscale image dataset. 115
Figure 43. Flowchart of CNN model testing on greyscale images dataset. 115
Figure 44. Training and validation performance of inception v3. 117
Figure 45. Crack detection performed on CPU using LabVIEW and Python. 119
Figure 46. Crack detection performed on CPU + TPU using LabVIEW and Python. 122
Figure 47. Sample images of concrete cube surface. 123
Figure 48. Result images after bilateral filter operation. 123
Figure 49. Result images after subtraction operation. 123
Figure 50. Result images after contrast stretching operation. 124
Figure 51. Result images after local threshold operation. 124
Figure 52. Testing images of surfaces with crack..126
Figure 53. Result of pre-processing applied on testing images. 126
Figure 54. Testing images of cracks for line emphasis method. 128
Figure 55. Result images of line emphasis method. 128
Figure 56. Testing images of cracks for percolation method...129
Figure 57. Result images of percolation method. 130
Figure 58. Result of classification using SVM. 132
Figure 59. Result of classification using KNN. 132
Figure 60. Learning curve of KNN. 133
Figure 61. Result images of crack detection in concrete cubes. 139
Figure 62. Sample images of concrete cubes and their results. 147

Figure 63. Concrete crack detection software interface..152
Figure 64. Crack monitoring graphs with respect to load...153
Figure 65. Flow of crack analysis...159
Figure 66. Sample images of cube -1 and cube -2 before and after compression test...........161
Figure 67. Crack and load monitoring graphs. 165
Figure 68. Concrete cube-A images showing the initial crack and final crack. 169
Figure 69. Progressive and individual crack monitoring with applied load for cube-A........170
Figure 70. Concrete cube-B images showing initial crack and final crack. 176
Figure 71. Progressive and individual crack monitoring with applied load for cube-B........177
Figure 72. Concrete cube-C images showing initial crack and final crack. 184
Figure 73. Progressive and individual crack monitoring with applied load for cube-C........185
Figure 74. Concrete cube-D images showing initial crack and final crack. 191
Figure 75. Progressive and individual crack monitoring with applied load for cube-D........192

List of Abbreviations

ACDM	Automatic Crack Detection and Measurement
ACM	Active Contour Model
AI	Artificial Intelligence
ANN	Artificial Neural Network
ASIC	Application-Specific Integrated Circuit
AUC	Area Under the receiver operating characteristic Curve
AWS	Amazon Web Service
BGC	Border Greyscale Chain
BMS	Bridge Management System
CCE	Cross Curvature Evaluation
CMOS	Complementary Metal Oxide Semiconductor
CNN	Convolutional Neural Network
CODEBRIM	COncrete DEfect BRIdge IMage
CPU	Central Processing Unit
CSV	Comma Separated Value
CV	Computer Vision
DAQ	Data Acquisition System
DHDV	Digital Highway Data Vehicle
DIC	Digital Image Corelation
DL	Deep Learning
FAS	Fourier Amplitude Spectra
FEM	Finite Element Method
FC	Fully Connected
FHWA	Federal High Way Administration
FOV	Field Of View

FPF	False Positive Fraction
FPR	False Positive Rate
GB	Giga Byte
GHz	Giga Hertz
GPR	Ground Penetrating Radar
GPU	Graphical Processing Unit
HDD	Hard Disk Drive
HMA	Hot-Mix Asphalt
HOG	Histogram of Oriented Gradients
HTF	Hough Transform based Feature
IE	Impact Echo
IRT	InfraRed Thermography
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
LBP	Local Binary Pattern
LSTM	Long Short Term Memory
LTPP	Long Term Pavement Performance
METU	Middle East Technical University
MF	Moment Features
ML	Machine Learning
MLC	Maximum Likelihood Classifier
MP	Mega Pixel
NCS	Neural Compute Stick
NDE	Non-Destructive Evaluation
NDT	Non-Destructive Testing
NI	National Instruments
PCA	Principal Component Analysis
PCC	Portland Cement Concrete

RAM	Random Access Memory
ROA	Region Of Aggregation
ROB	Region Of Belief
ROC	Receiver Operating Characteristic
ROI	Region Of Interest
RTU	Remote Terminal Unit
SCC	Self-Compacting Concrete
SIFT	Scale Invariant Feature Transform
SMASH	System, Material And Structure Health
SSD	Solid State Drive
SVM	Support Vector Machine
TB	Tera Byte
TOPS	Trillion Operations Per Second
TPF	True Positive Fraction
TPR	True Positive Rate
TPU	Tensor Processing Unit
UAV	Unmanned Aerial Vehicle
UPV	Ultrasonic Pulse Velocity
USU	Utah State University

Contents

Certificate iv

Declaration v

Certificate vi

Acknowledgement vii

Abstract ix

Contents xii

List of Tables xv

List of Figures xvii

List of Abbreviations xx

Chapter 1 Introduction 23

1.1 Overview 23

1.2 Computer vision in civil engineering 25

1.3 Need of the study 28

1.4 Objectives 32

1.5 Scope of work 33

1.6 Organization of thesis 33

Chapter 2 Literature review 36

2.1 Overview 36

2.2 Crack detection methodologies 37

2.3 Image processing algorithms 40

2.4 Artificial intelligence based approaches 47

2.5 Crack properties measurement 53

2.6 Indian knowledge system 57

2.7 Summary 60

Chapter 3 Experimental setup 62

3.1 Requirement study 62

3.2 Hardware selection 64

3.3 Software selection 71

3.4 System integration 72

3.5 Concrete cube images and load data 77

Chapter 4 Crack detection methodology 83

4.1 Research methodology 83

4.2 Image pre-processing 86

4.3 Conventional image processing 88

4.3.1 Method 1 - Subtraction, line emphasis and iterative threshold 88

4.3.2 Method 2 - Percolation based crack detection 91

4.4 Machine learning 94

4.5 Convolutional neural network 97

4.5.1 Overview 98

4.5.2 Performance metrics 99

4.5.3 Baseline model selection 101

4.5.4 Dataset preparation 102

4.5.4.1 Binary contour image dataset 106

4.5.4.2 Greyscale contour image dataset 109

4.5.5 Model training and testing framework 112

4.5.6 Real-time implementation approach 118

4.6 Results and discussions 122

4.6.1 Image pre-processing 122

4.6.2 Conventional image processing 127

4.6.2.1 Method 1 - Subtraction, line emphasis and iterative threshold 127

4.6.2.2 Method 2 – Percolation processing 129

4.6.3 Machine learning 130

4.6.4 Performance evaluation of convolutional neural network 135

4.6.4.1 Binary contour images 136
4.6.4.2 Greyscale contour images 139
4.6.4.3 Comparative assessment of implemented methods 147
4.6.4.4 Comparison with methods presented in the literature 149
4.6.5 Software implementation 151
4.6.6 Results of real-time implementation 153
4.6.7 Summary 155
Chapter 5 Crack-load monitoring and analysis 157
5.1 Overview 157
5.2 Measurement of crack length and crack area 158
5.3 Analysis of crack properties 160
5.4 Analysis of crack-load data 166
5.4.1 Concrete cube - A 166
5.4.2 Concrete cube - B 173
5.4.3 Concrete cube – C 180
5.4.4 Concrete cube – D 189
Chapter 6 Conclusion and Future scope 196
6.1 Conclusion 196
6.2 Future scope 198
List of Publications 200
Journal Publications 200
Conference Publications 200
Book chapter Publications 201
References 202

Abstract

Reinforced concrete is widely used for the construction of bridges, buildings, highways, dams, power plants and many other infrastructures. Monitoring the structural health of these infrastructures is essential for their uninterrupted functioning and assessing their deterioration due to loading and environmental factors. The monitoring also helps in the estimation of its load-carrying capacity, serviceability and need for repair. Cracking in concrete structures is one of the critical parameters representing the health of the structure. Concrete cracking occurs due to many reasons like shrinkage, heaving, premature drying, and excessive loading which leads to a reduction in the strength of structures. Trained personnel monitor the development of cracks and their progression at the critical locations of the structures through a physical vision at regular intervals of time.

Structures like bridges, buildings, roads, tunnels, historical monuments and many others are inspected at regular intervals to estimate their deterioration and to prevent further accidents which may directly affect human life. Different methods like acoustic, ultrasonic and image processing-based inspection methods have been deployed to carry out an assessment of such concrete structures. Physical inspection of structures for health monitoring is time-consuming, costly and risky. Automatic detection of cracks in concrete structures of various shapes and scales holds worthy importance in the area of structural health monitoring. Advances in image acquisition, processing techniques, and computational resources have made vision systems, a cost-effective and accurate technique for structural health assessment.

The present work is aimed to address the concrete crack detection problem by developing a novel system using machine vision and deep learning. The work covers concrete crack monitoring by identifying the location of the crack, the number of cracks, the length of the cracks, and the area of the cracks. The methodology is applied for crack monitoring of concrete cube of standard dimensions 150 mm × 150 mm × 150 mm subjected to compressive loading. An innovative system has been developed by integrating machine vision and convolutional neural networks to acquire real-time images of concrete surfaces, detect concrete cracks, and extract parameters related to cracks, such as the number of cracks, location, length, and area, in synchronization with the applied compressive loading. The present system is implemented to capture 1 sample of image and load data per second, for crack monitoring during compression testing of concrete cubes of size 150 mm × 150 mm × 150 mm. The crack detection methodology presented in the work offers a better understanding of concrete cube behaviour with respect to applied compression load.

An integrated experimental setup is developed to acquire consistent images of the concrete cube along with the applied load during the compression testing of the cube. The setup comprises of a state-of-the-art industrial machine vision system, load cell and load cell data acquisition system. Several concrete cubes are tested for compressive load and surface images of the cubes are acquired. Due to random unevenness on the surface of concrete blocks, designing an efficient crack detection algorithm becomes challenging. The problem of crack detection is first addressed using subtraction, line emphasis and iterative threshold-based method. Additional method based on percolation as well as machine learning are also applied for the crack detection. Each of these methods lacks in meeting expected performance in terms of accuracy and precision in crack detection.

Convolutional neural network based methodology has been applied to overcome the challenges of variations in the concrete cube surface images, and similarity in the intensity level of dents, noises and spots present on the cube surface. With the use of the developed experimental setup, dataset collection and preparation are carried out for binary contour images and greyscale contour images of cracks and non-cracks. Initially, the neural network based method is used with a binary image dataset. The results obtained with the binary dataset are not satisfactory and hence greyscale data is used thereafter. But due to data imbalance between the crack and non-crack contours, the performance of the network with greyscale images do not have good recall and precision. Finally, the network is trained with a balanced and augmented greyscale image dataset to achieve satisfactory results in crack classification. Despite the challenges in accurate crack identification on uneven surface of concrete cube, the novel methodology shows improved performance in crack detection.

The Inception v3 model is having lower error rate, faster convergence and reduced computation time for object identification. Therefore, Inception v3 architecture is trained to detect the cracks from the image of concrete cube surface images in the most accurate manner. The Inceptionv3 model is trained and validated using more than 80,000 crack and 80,000 non-crack images dataset prepared manually using the concrete cube surface images. Popular data augmentation techniques are used to generate the training dataset. An average of 97.49% accuracy and 7.38% cross-entropy are achieved in the training whereas 97.67% accuracy and 7.69% cross-entropy

are achieved in the model validation. The training is carried out with a batch size of 100 and 5,000 epochs. An average accuracy of 99% has been achieved during the performance evaluation of crack detection on concrete cubes as presented in the results. The average values of precision, recall and F – score is obtained as 0.88, 0.98 and 0.93 respectively.

The performance of the proposed methodology is compared with the other strategies reported in the literature based on image acquisition device, infrastructure and CNN model. In comparison, it is observed that the performance of the present methodology is 2-10% better in terms of crack detection accuracy. In addition to that, the proposed methodology is also compared with other methods having similar approach like machine learning based crack detection, crack classification using the binary dataset, imbalance greyscale dataset and balanced greyscale dataset. The comparison justified the use of data augmentation techniques and a greyscale contour dataset prepared for accurate crack detection. The outcome of the machine vision system in graphical form is presented for various parameters of cracks like the number, location, length, the area with respect to compressive load for different concrete cubes. Comprehensive crack-load monitoring and analysis is presented for multiple concrete cubes.

A detailed crack-load analysis data is presented which comprises of analysis of individual crack-related data, crack-highlighted images of concrete cubes, load versus progression in the crack length, load versus progression in the crack area as well as cumulative crack length and crack area with respect to load for the concrete cube. The observations obtained from the crack load analysis can provide vital information and improve the understanding of concrete behaviour subjected to loading. The developed machine vision and deep learning based method is a step forward in the structural health monitoring of real-life concrete structures like buildings, bridges, and pavements. The machine vision system can be implemented on different concrete structures for acquiring real-time data on crack development and progression. The developed framework will be an effective tool for engineers working in the domain of structural health monitoring of concrete structures.

Chapter 1 Introduction

Chapter 1 of the thesis covers the introduction to the area of structural health monitoring with a specific focus on concrete crack detection. The areas like concrete mix design, damage assessment of structures, the importance of concrete testing, vision based crack detection and its advantages over manual inspection are briefly covered. An introduction to the area of computer vision and its applications in civil engineering are discussed in detail. The need for the study, objectives and the scope of work is covered in the later part of the chapter followed by the organization of the entire thesis.

1.1 Overview

Concrete mix is one of the most widely used construction materials in the world. The mix includes crushed stones, gravel, and sand which are typically bound together with cement and water. The proportion of each component in concrete is determined based on the required properties for the construction work. The mix proportions are either nominal mix or design mix. The nominal mix of concrete is used for ordinary construction work such as a small residential structure. On the other hand, the design mix relies on the proportions finalized using laboratory tests which are performed to determine the compressive strength of the mixture. Additionally, there are various types of concrete like plain concrete, reinforced concrete, precast concrete, high-density concrete, ready mix concrete, decorative concrete and rapid set concrete (Different types of concrete , 2020) (Raj , 2017). The varieties have grown into numbers due to the different requirements and applications, for example, ready-mix concrete is widely popular since it is a machine mix which has higher precision and large volume readily available at the construction site. Reinforced concrete is widely used for the construction of infrastructure projects like bridges, buildings, highways, dams, and power plants. Annually, billions of tons of concrete are used around the world for the construction of various types of structures.

Various additional fibres and admixtures are embedded in the concrete to improve various properties of concrete necessary in different construction applications.

However, concrete remains a major ingredient of construction work and is almost used twice as much in construction as all other building materials.

Numerous old construction work and even relatively newer structures collapse due to various natural and other causes. The failure results when the structure is stressed beyond its strength limit. There are various types of failures like a catastrophic failure, failure from fatigue or corrosion, failures caused by manufacturing errors, and the use of defective materials that can also result in failure. In a well-engineered structure, even a localized failure may not cause immediate or progressive collapse. The news of collapses in bridges, buildings, and walkways are very common nowadays. These failures and collapse can result in the loss of human life and has huge social as well as economic impact. Damage to a nuclear plant or a nuclear vessel may create a leakage and a hazardous situation. As a result, professionals, engineers and researchers see immense potential in the health monitoring of new as well old structures and construction work (Shankar H. Sanni , 2018). There can be several causes for the failure of concrete and monitoring the causes or the indicators of structural health is essential for their uninterrupted functioning.

Cracking in concrete structures is one of the most important and primary indicators of a structure’s health. It indicates the deterioration in the strength of the structure and warns against possible failure. Generally, physical inspections are carried out to detect defects in structures for further rectification, repair and reinforcement as and when required. The physical inspection of cracks in concrete structures of various sizes can assist in determining the overall serviceability of the structure.

Image based automated or semi-automated detection of cracks has the potential to overcome the limitations of manual inspection. In alternative to physical inspection, images of critical locations of the structure can be captured by state-of-the-art image acquisition devices. The processing of images provides information about the current condition of structures. With recent advancements in computational algorithms, vision-based inspection is emerging as an efficient technique for monitoring the structural health of structures. Using the latest Artificial Intelligence (AI) based techniques, a better interpretation of collected data can be obtained automatically.

These advanced techniques can also provide the ability to predict the residual strength of structural elements.

Cracks in concrete structures are the only external and primary indication of deformities developing in the structure. Early and accurate identification of these cracks is highly beneficial in determining the extent of damage and serviceability of the structure (W. O. Ajagbe , 2018).

The assessment of structures like bridges, tunnels, pavements, roads, pipes, buildings, monuments, and historical sites is periodically carried out via physical inspection. Field engineers visit these sites and acquire various photographs to highlight and evaluate the structure's health. These photographs are visualized by human experts and an assessment of the structure is carried out. It is a highly expensive, discontinuous and cumbersome process of evaluating a structure. This conventional approach of assessment can be automated using vision based intelligent system. The majority of automated inspection applications are carried out by image processing techniques. These inspection tasks required good-quality image acquisition devices, computation systems and intelligent algorithms. The advancement of technology has enabled professionals across the globe to address the problem of crack identification. Automated and vision-based monitoring of concrete surfaces for accurate crack identification has emerged as a cost-effective, faster and reliable solution in comparison with a physical inspection.

1.2 Computer vision in civil engineering

Computer Vision (CV) refers to a specific field that can process images and videos to extract meaningful information. According to IBM, computer vision is the field of AI that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs and take actions or make recommendations based on that information (computer-vision , 2022). The field primarily involves the use of different kinds of image acquisition devices along with image processing techniques to extract meaningful information from images and/or videos. An image acquisition device and computation resources are two primary hardware required for a computer vision application. Software is built to deploy on the computation resources and provide real-time results which may be in form of actions, suggestions or communication. The software uses conventional image processing-based techniques in order to process the image and/or video. If the application demands, Machine Learning (ML)and/or Deep Learning (DL) based methodologies can be integrated with the conventional image processing-based techniques.

Various applications of computer vision include gesture recognition (Cipolla Roberto , 1998), autonomous cars (Hussain Rasheed , 2019), face recognition (David White , 2015), and machine vision (Steger Carsten , 2018). Numerous operations, algorithms and methods are available in computer vision domain to extract meaningful information from images and videos. The applications of computer vision cover a wide spectrum of engineering, science, agriculture and technology. With the advancement of technology, the prices of good quality cameras and computing devices have decreased to a greater extent. It has enabled people to explore new applications of computer vision and improve the performance of the existing systems.

Computer vision has potential to address different applications of inspection and damage assessment in civil engineering. The applications can be categorized into two types which are inspection and monitoring (Dong , 2021). Both categories of applications cover aspects such as the identification of structural components, characterizing local and global visible damage, acquiring, storing and communicating structural images and detecting changes from a reference image (Billie F Spencer Jr , 2019). Throughout the life cycle of civil assets, construction, operation and maintenance phases require monitoring to assure reasonable decision makings. Each of the phases can be addressed using computer vision applications (S. Xua , 2019). Based on the fact that crack pixels in pavement images had distinct grayscale intensities compared to their surrounding non-crack pixels, JaChing Chou et. al. proposed a pavement crack detection algorithm based on fuzzy logic (JaChing Chou , 1994).

The concept of Finite Element Method (FEM) was applied by Sun et. al. (Sun , 2005) to determine the complete, two-dimensional displacement field during the image correlation process on digital images. The authors used both numerical studies and a real experiment to verify the proposed formulation and showed that the image correlation with the finite element formulation is computationally efficient, accurate, and robust. Hung et. al. developed a fast and simple detection algorithm based on DIC for measurement of the surface deformation of planar objects (Hung , 2003). Besnard et. al. (Besnard , 2006) introduced the concept of multiscale approach on top of FEM based DIC method to generate meaningful solution for a fine texture and large initial displacement measurement. Sylvie Chambon et. al. (Sylvie Chambon , 2009) proposed a new approach to image-based crack detection, GaMM, based on a multi-scale extraction and Markovian segmentation, which reduced the percentage of false positives in comparison to morphological methods that combine thresholding and refinement by morphological analysis.

Researchers have applied the use of computer vision in inspections, and gauging applications pertaining to civil engineering, especially construction work. If we consider a practical application of building inspection, the biggest limitation faced is the limited field of view of the good quality industrial camera. However, inspections and measurements for civil engineering using image processing can be performed for laboratory scale objects.

An automatic crack detection system was proposed by Jinlong Chen et. al. (Jinlong Chen , 2010) by employing a coarse-to-fine methodology that also included concepts like Region Of Aggregation (ROA) and Region Of Belief (ROB) for segmentation and localization of cracks. Ying et. al. (Ying , 2010) proposed a beamlet transform-based technique for pavement crack detection and classification, which was more robust in extracting linear features in the presence of noise. In 2011, Tuong Lam Nguyen et. al. (Tuong Lam Nguyen , 2011) developed a new automated method of fracture identification and quantification based on standard DIC approach.

Henrique Oliveira et. al. proposed an integrated system, CrackIT, for automatic detection and characterization of cracks in flexible pavement surfaces using a combination of unsupervised learning followed by supervised learning, thus eliminating the need for manually labelling the samples (Henrique Oliveira , 2013). It was noted that although CrackIT was able to detect multiple cracks in the same image, it had difficulty in dealing with cracks less than 2 mm width.

Danfeng Xie et. al. demonstrated the potential for Deep Learning based pavement crack detection by applying ConvNets on a dataset of 500 images of size 3264 × 2448, collected using a low-cost smartphone (Danfeng Xie , 2017). Chen Feng et. al. proposed a deep active learning strategy for civil infrastructure defect detection and classification, where they used a deep residual network called ResNet to train a small set of images with defect labels and use this low-accuracy defect detector to filter out many non-defect images (Chen Feng , 2017). Kumar et. al. used a dual channel convolutional neural network model for crack detection in concrete (Kumar , 2020). The primary network used was a single-channel CNN in which feature extraction was done using intermediate max-pooling layers. The model was trained with a learning rate of 0.0005 and achieved an accuracy of 90.5%.

Thus, computer vision has been widely used and applied for addressing various applications in civil engineering. Now, with the use of AI and ML, researchers are finding solutions to much complex and challenging applications which were difficult to address using conventional methods.

1.3 Need of the study

It is important to understand the concept and behaviour of different types of concrete to use them in practical applications. Before taking them into an application, it is important to assess their effectiveness by casting and testing them in the laboratory. Various tests like compression tests, tensile tests and others are carried out on concrete elements to understand their behaviour under loading. Measurements like load, strain, and displacement are carried out with the help of different techniques. Various parameters such as compressive strength, flexural strength, the strain developed, displacements, temperature profile, crack initiation, crack propagation as well as depth and length of cracks are evaluated during the testing of the concrete element.

Non-contact measurement of concrete elements holds importance in the study of the behaviour of various concrete elements tested under different kinds of testing. The need for more accurate measurement of the above-mentioned parameters drives the development of automated vision-based set-up. It is very difficult to manually locate the crack initiation and propagation with respect to force in real time during a testing experiment (Talaat , 2021). Numerous practical challenges are faced in the manual inspection of concrete structures. For a bridge inspection, a certain level of human expertise is required along with the time and cost involved. Further, the process of manual inspection is a periodic procedure instead of continuous monitoring.

Various automatic methods have been devised to measure parameters like crack, vibration, strain, displacement, and load at critical locations in concrete structures. Other disadvantages include cost, time, a requirement for human expertise, numerical data collection and analysis which make such systems less popular (Eleni N Chatzi , 2011).Images of cracks on different surfaces are presented in Figure 1. The variations just give an idea about the problem under consideration.

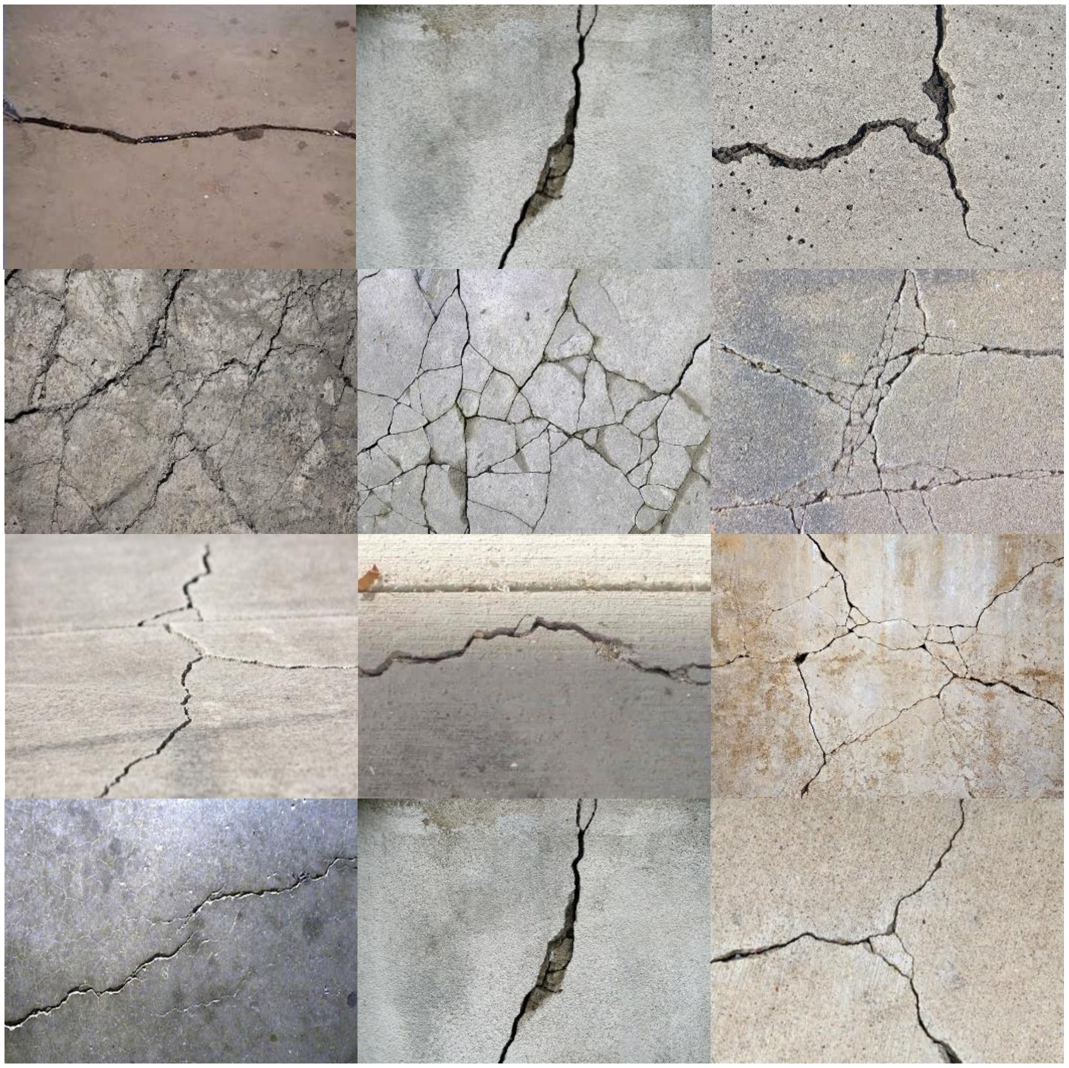


Figure 1. Sample concrete crack images.

Conventionally, sketches of cracks are manually drawn to understand the behaviour of a concrete element. This requires good knowledge and experience in concrete testing.

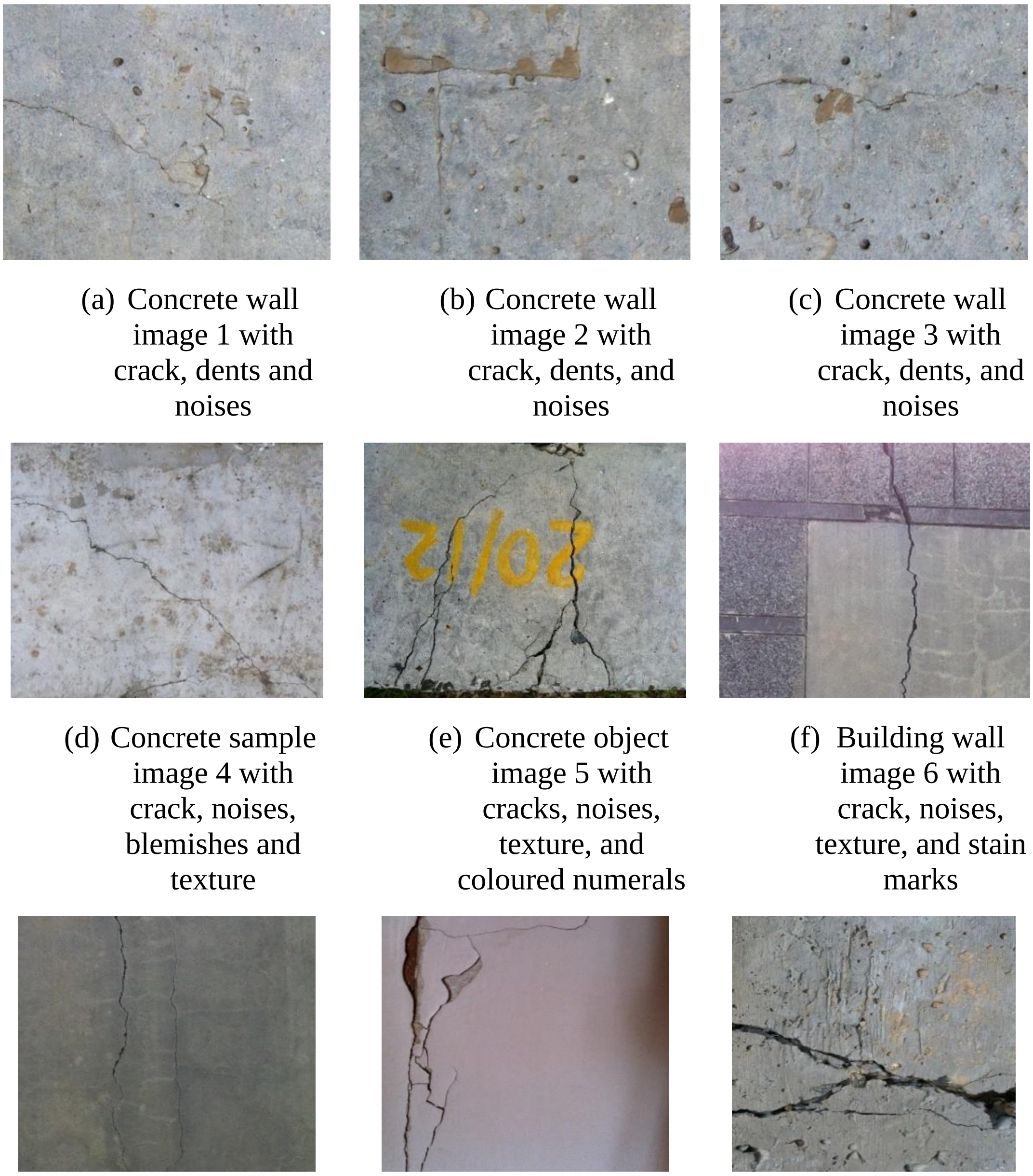
The practical limitation of manual methods is time consuming; the need of technical expertise, cost, and point-to-point measurement hinders their continuous usage for practical crack detection on concrete surfaces. Also, it is impossible to measure the width of micro cracks with utmost accuracy and precision during the experiments in real-time as well as at the end of experiments. Several manual tools and techniques are available to facilitate the crack properties measurements such as concrete crack gauge, crack width gauge and similar other gauges (Construction Materials Testing Equipment , 2016).

There is a need to inspect concrete elements during testing and obtain important parameters like crack length, width, depth, crack initiation, and number of cracks with the application of force online during the experiment. Accurate inspection and measurement of the crack properties would produce reliable results and would assist in much better research and development of concrete technology and testing. Concrete crack has certain external features which are listed below (Hongke Xu , 2013). It is important to consider them to understand the problem in more detail.

External features:

1. The crack is a comparatively darker area than the background. Its pixel value is relatively low.
2. It may be in a discrete form or a continuous form.
3. For a single image, the area of the crack would be mostly larger than the noises, dents, interference objects and smaller than the background area.
4. The growth or propagation of cracks on a concrete surface or any other surface is purely random.
5. Noises/dents/stain marks are present on the surface and its position and growth are random too.

The discrete form mentioned here represents the cracks on the concrete surface which are discontinuous and seen in small patches. With the increase in load, the crack previously seen in small patches joins together and appears as a continuous and long crack.



(a) Concrete wall image 1 with crack, dents and noises

(b) Concrete wall image 2 with crack, dents, and noises

(c) Concrete wall image 3 with crack, dents, and noises

(d) Concrete sample image 4 with crack, noises, blemishes and texture

(e) Concrete object image 5 with cracks, noises, texture, and coloured numerals

(f) Building wall image 6 with crack, noises, texture, and stain marks

(g) Concrete object image 7 with cracks, and texture

(h) Concrete wall image 8 with cracks, texture, shadow and stain marks

(i) Concrete wall image 9 with cracks, dents, texture, and stain marks

Figure 2 Samples of cracks in concrete with variations.

Images of various concrete specimens and structures have variations and issues like noises, dents, blemishes, stain marks, texture on the concrete surface, shadow, blur, and non-uniform illumination effects at the time of image acquisition, and variations in the shades of different civil structures like concrete structures, roads, and asphalt pavements.

Figure 2 shows images of concrete surface with all these variations. These varieties of issues pose a great challenge in accurate crack detection. And it is the main motivation of researchers trying to explore the domain and design algorithms for accurate and robust crack detection.

Characteristics of concrete structures are affected by an inferior standard of maintenance, fluctuation of temperature, and an excessive amount of load. Cracks are the only external and primary indication of deformities which need to be inspected at earlier stages for the strength and reliability of the concrete structure. Various kind of cracks are to be identified in distinct types of surfaces like bridges, building, tunnel, road, and pavement. Examination of cracks in practical bases is requisite in a nuclear plant to find out any leakages to prevent a hazardous explosion. Automatic crack detection is severely affected by irregular conditions in illumination, shading, and dents in the images of the concrete structure. Abstruse to discriminate the difference between the crack part, noise and non-crack part of the concrete surface. The motive behind this work is to automate and digitize the concrete cube testing process using the power of vision and computation.

1.4 Objectives

The overall objective of the work is to apply the concepts of machine vision to carry out accurate crack detection and perform crack-load analysis. The objectives of the work are:

1. To develop a CNN based accurate and robust algorithm to perform crack detection on surface images of concrete cubes.
2. To acquire synchronized data of concrete cube surface image and applied compressive load to carry out monitoring of crack initiation and propagation during the compression testing.
3. To extract the number of cracks, crack length, and crack area with respect to applied load for different strengths of concrete cube mix design.
4. To perform crack-load monitoring and analysis.

1.5 Scope of work

The scope of the work, to achieve the objectives, are:

1. To design and develop an indigenous and synchronized image and load data acquisition setup compatible with the compression testing machine.
2. To develop a software algorithm integrated with the data acquisition setup to perform synchronized data acquisition, real-time processing, CNN-based crack detection, visualization of crack detection and data logging.
3. To cast various concrete cubes of size 150 mm × 150 mm × 150 mm with different compressive strengths and perform compression tests as well as simultaneous crack detection and data logging.
4. To perform crack-load analysis using the acquired data and find out the correlation between crack properties and loading behaviour.

1.6 Organization of thesis

The thesis is organized properly into six chapters. The introduction chapter comprises the primary overview of the camera based or vision-based inspection systems researched for damage assessment of various types of concrete structures. A brief overview of computer vision applications in the area of civil engineering is stated to understand the importance of the domain. The sub-area of crack detection faces different issues and challenges which are associated with computer vision-based techniques. The discussions are kept as the need for the study. The objectives of the work and the scope of the work are also mentioned in the chapter.

After the introduction chapter, a brief review of the literature work is presented in chapter 2. The chapter covers a comprehensive summary of various crack detection methodologies including the notable contributions, their assumptions, specifications, structure type and other related parameters. Conventional image processing and computer vision-based techniques are discussed in depth. Advanced methods which are based on the use of AI, ML and DL are discussed. Crack detection and damage assessment of different concrete and other surfaces using the advantages of these advanced methods are also reported in the chapter. Several authors have proposed different methods and approaches for crack properties measurement. Properties like crack length, crack area, crack orientation, and crack width are extremely important and crucial parameters for damage assessment. Notable contributions in the domain of crack properties measurement and estimation are also discussed in detail. The work presented in the area considering the Indian researchers is also discussed in the Indian knowledge system. The chapter ends with a brief summary and observations of the literature reported.

Chapter 3 discussed the development of the experimental setup. The setup is developed to acquire real-time data during the compression test in a synchronous manner. A detailed requirement study is discussed in detail. The hardware and software used to develop the experimental setup are discussed in depth along with the specifications of each component. Continuous data of surface images and applied compression load are acquired using the experimentation setup. Few sample images and load data are kept in the chapter to verify the results of the data acquisition process. The details of the compression testing machine, sample cube preparation and testing methodology are discussed. Convolutional Neural Networks (CNN) require efficient computation resources to operate in real-time and perform model training, testing and deployment. National Instrument’s (NI) Laboratory virtual instrument engineer’s workbench (LabVIEW) is used as primary frontend software to implement the overall methodology. CNN frameworks and libraries which are widely popular in python are discussed along with debugging and troubleshooting support and are well managed for python. The chapter also covers the integration aspects of the setup which comprises control and flow of data between multiple hardware and software.

Chapter 4 introduces the proposed methodology which is implemented for accurate and robust crack detection on surface images of concrete cubes. It covers the image pre-processing facets of the overall methodology followed by trials of crack detection performed using multiple strategies in conventional image processing way. ML based method comprising manual feature extraction, data preparation, supervised training, testing and analysis is also discussed. The chapter also includes a brief introduction to a CNN and various approaches used for concrete crack detection.

The approaches include work carried out using the binary dataset, greyscale data with an imbalance between classes and greyscale dataset with data augmentation and balanced class data. Different layers of CNN are elaborated on in the chapter. Several approaches which are conventionally used for dataset generation are discussed. Also, popular models like Alexnet, RESNET, VGG16, and Inception are compared along with their architecture. The model selection strategy is discussed to justify the use of the Inception v3 network. In the results and discussion sub-section of the chapter results of each step are discussed in brief along with necessary images, graphs, and data in tabular form.

Chapter 5 covers the analysis of the accumulated crack-load synchronized data. The data was acquired for concrete cubes with different strengths and grades. Along with crack and load information, measurement of crack properties like crack length, crack area, and the number of cracks with respect to applied compression load is presented in the chapter. The obtained data provided insightful information on concrete cube behaviour under compression load. The properties measured for each crack detected on the concrete cube surface images provide the individual crack growth and propagation with respect to load. The data acquisition, crack detection, monitoring and analysis are presented for various concrete cubes. The crack detection results are shown with images along with quantitative and qualitative analysis information represented in the tables. The chapter also includes a brief conclusion of the monitoring, analysis and results obtained.

Chapter 6 includes a brief conclusion of the overall contribution of the work. Further work which can be carried out in the area is covered in the future scope. The thesis also includes a list of publications and references cited in the thesis.

Chapter 2 Literature review

Chapter 2 of the thesis covers a detailed literature review related to crack detection in concrete structures and other infrastructures. Crack detection, measurement and estimation of crack properties like crack length, crack area, crack width, crack depth, location of the crack, orientation of crack, damage assessment methods, and other associated sub-domains in the area of structural health monitoring were worked upon by the research community. AI and ML based crack detection and damage assessment methods have gained popularity in recent years. CNN-based crack detection methods are also discussed in the review. Relevant and novel contributions are discussed in detail along with assumptions, performance and limitations of each work. A review summary is presented at the end of the chapter.

2.1 Overview

Concrete is the most widely used construction material because of its strong, durable, heat-resistant and water-resistant properties. Many natural and manmade factors adversely affect the serviceability of a concrete structure. Structural damage may easily occur in bridges, buildings, pavements, roads, tunnels, pipes, water tanks, and nuclear vessels. The damage may further lead to fracture or collapse and may cause serious safety threats to society. The damages are conventionally seen in form of surface cracks, which should be detected to extend the service life of concrete engineering structures. Cracks in concrete structures can initiate at any time during the lifetime of the structure. These cracks can occur due to plastic shrinkage, constructional movements, overloading concrete elements, or creep (Safiuddin , 2018).

Chemical reactions such as alkali-aggregate reactions and corrosion can also induce cracking in concrete structures. Cracks being one of the first and most important indicators of a concrete structure's health, its accurate and continuous detection can be advantageous in many ways. These cracks greatly reduce the load-bearing capacity, waterproofing performance and durability of the structure. If cracks are detected accurately and timely, it may cause serious safety concerns to human life.

Concrete elements/structures for which automatic crack detection using computer vision have been explored are mentioned below (Jianghua Deng , 2022).

1. Concrete walls of a building
2. Bridges
3. Heritage sites and historic monuments
4. Underground sewage lines/pipes
5. Nuclear vessels and structures
6. Concrete pavements
7. Asphalt pavements
8. Roads
9. Tunnels
10. Water tanks

A concrete crack is a linear fracture which extends partly or completely through a structural member. Visual monitoring of concrete structures is a conventional approach to assessment. It is the first primary step toward understanding the nature of cracks and their underlying causes. During the visual inspection, crack length, crack area, crack width and crack depth are assessed. Cracks with rust marks on the surface can be a result of steel corrosion cracks. Another example can be that of inclined cracks over concrete beams near the support which occurred due to shear stress. Crack length and crack area can be monitored to provide the progression of the crack over a time period. Crack width provides information on the severity of the cracks whereas crack depth is used to evaluate the overall structural integrity of the structure or concrete element. Notable contributions in these domains are discussed in brief.

2.2 Crack detection methodologies

Crack evaluation should be carried out based on these crack properties both in real-life structures as well as in laboratory environments. The study of crack properties in real-life structures can enable structural engineers to take appropriate action based on the extent of the damage. In the case of a laboratory environment, the properties can provide detailed and progressive information on concrete performance in different concrete testing with respect to applied loading. It will also enable the engineers to develop better concrete. Crack length is conventionally measured with the use of standard measuring tapes. However, the crack will never be straight and hence measuring the crack length using a standard tape may be inaccurate.

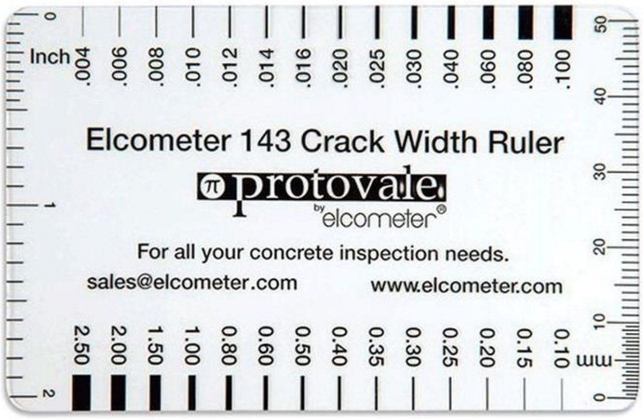


Figure 3. Crack width ruler (elcometer-143-crack-width-ruler , 2022).

A crack width ruler or crack gauge is a widely used tool to estimate crack width. Cracks can be classified as hairline, thin or severe depending on their width in millimetres (Gurmu , 2021). Conventionally, the crack-width ruler is placed on top of the crack. The ruler is then adjusted to match the actual crack width and the crack width on the ruler. Figure 3 shows a typical crack width ruler which is typically used for the manual measurement of crack width. The rules have scale markings starting from 0.1 mm to 2.5 mm and 0.004 inches to 0.1 inches. Figure 4 shows the different tools widely used for the measurement of crack width (Concrete Cracking: Evaluating width, depth and movement , 2022). Different methods comprise crack comparator cards, crack width gauges, digital callipers, optical comparators, and crack width microscopes.

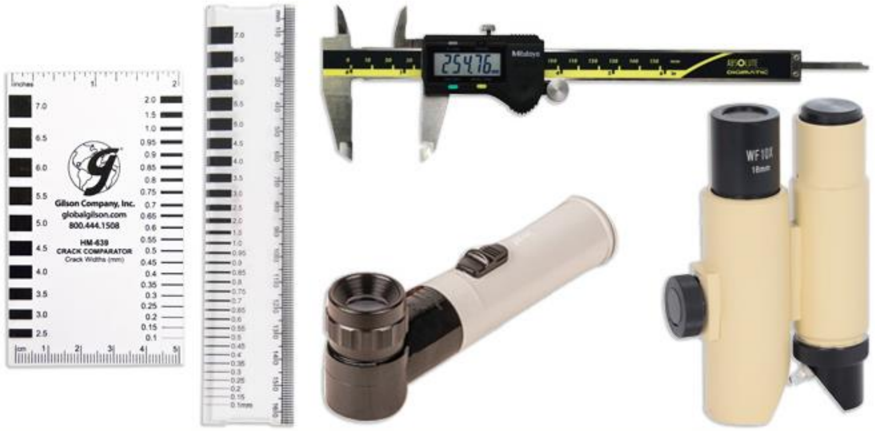


Figure 4. Tools to measure crack width.

Similarly, crack depth gauges are also available using which the depth of a crack can be measured. These gauges are tapered and in form of a scale which can be easily used by engineers to evaluate the crack depth in a concrete member. Another method widely used for crack depth measurement is an impact-echo test. The test comprises two transducers which are placed on the concrete surface to assess the crack depth. A stress pulse is generated at the element surface of one transmitting transducer which spreads into the test object and gets reflected by cracks, flaws or interfaces or boundaries. Another receiving transducer received a surface response caused by the arrival of the waves. The received data is analysed in the frequency domain to measure the wave speed and the thickness (V. Mohan Malhotra , 2003).

Ultrasonic Pulse Velocity (UPV) is an effective Non-Destructive Testing (NDT) method normally used for the quality control of concrete materials, and damage detection in structural members. The UPV methods have traditionally been used for the quality control of homogeneous materials such as metals and welded connections. Owing to the recent advancement in transducer technology, the test has been widely accepted in the testing of concrete members (3 Methods for Crack Depth Measurement in Concrete , 2017). Looking into the disadvantages of manual inspection methods, an automatic, continuous and accurate approach is explored in recent years by researchers. The approach comprises of using the power of vision system and image processing based intelligent algorithms applied for crack detection and properties measurement. In recent years, image-based crack detection has received a lot of attention. Applications of image processing to monitor structure health has matured and offers non-contact measurement, good amount area, mm accuracy in large structures and micron accuracy in laboratory scale structures.

Hafiz Sulaiman Munawar et. al. provided a detailed review of image processing methods used for crack detection (Hafiz Sulaiman Munawar , 2021). The authors reviewed articles in two categories which are image processing-based crack detection and ML based crack detection. A total of 30 articles from both categories were studied, compared and presented. Out of the 30 articles, 67% of the articles focussed on image processing methods while 33% of the articles used ML techniques. Recursive tree edge pruning, Gabor filter, particle filter, beamlet transform, canny edge detection, hessian matrix, and Shi-Tomasi feature point detection processing methods were utilized to perform crack detection. These crack detection methods were studied on pavement, bridges, concrete structures, and noisy concrete surfaces.

Digital cameras, IP cameras, Canon IXUS 80, Sony CyberShot DSC-F828, UAV cameras, DJI Mavic pro, and consumer-grade digital cameras were utilized for image acquisition. The result in terms of accuracy, precision, recall, Area Under the Receiver Operating Characteristics (ROC) Curve (AUC), and error range, was evaluated along with their limitations for each of the studied methods. Similarly, ML based crack detection methods, the surface used for the study, the image acquisition device, results and limitations were presented in the study. K means clustering, support vector machines, random forest, logistic regression, deep convolution encoder-decoder network, and various other convolution neural network based methods were studied.

The authors concluded that the majority of the articles focussed on crack detection only. The methods showed good performance in terms of crack detection where the precision values range between 75%-100%. However, dimension calculation of the crack was not performed in the majority of the work. The authors suggested that more research should be carried out to focus on crack measurement and predict the severity of the crack using intelligent algorithms. In the years 2016-2022, most of the authors have used AI and ML based methods instead of conventional image processing-based algorithms.

2.3 Image processing algorithms

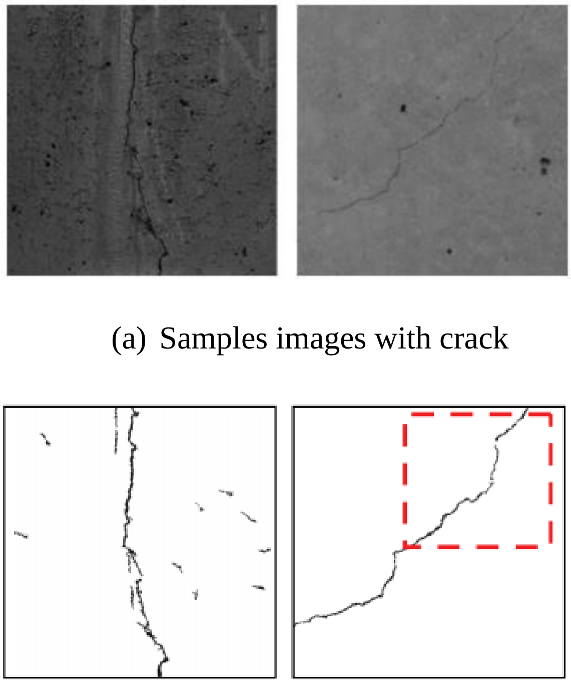
Image processing-based algorithms addressed to the improvements of crack detection on different types of concrete elements and structures have been explored by researchers. Work on algorithms to measure crack properties like length, width, and the orientation of cracks developed on concrete surfaces have been carried out (Bang Yeon Lee , 2013) (Tomoyuki Yamaguchi , 2008) (Yusuke Fujita , 2006). Surface defects on the concrete surface, crack depth measurement, and thermal behaviour of concrete structures are some of the few related subdomains which were explored by the researchers. Mainly they have tried to formulate different algorithms and methods to identify cracks on the surfaces of various concrete elements.

Uneven concrete surface, random noise on the surface, voids, dents, colour variations, and stain marks make it difficult to identify or formulate an algorithm which can accurately locate cracks and further carry out the damage assessment of the structure under observation.

Numerous researchers have reviewed the work carried out in the domain of crack detection using computer vision or imaging science (Christian Koch , 2015) (Ayaha Miyamoto , 2007) (Tran Hiep Dinh , 2016) (Arabinda Sharma , 2016) (Weiguo Lin , 2019). Concrete structures for which automatic crack detection using computer vision have been explored are walls and exterior as well as interior parts of a building, heritage sites and historic monuments, underground sewage lines and pipes, nuclear vessels and structures, asphalt pavements, roads and other infrastructure.

A literature study of the work carried out in the measurement and estimation of various crack properties is discussed here separately. Image processing-based algorithms addressed the improvements of crack detection on different types of concrete elements and structures been explored by researchers. A brief literature review is presented below to understand the work done in the area.

Tomoyuki Yamaguchi et.al. presented a novel approach to detecting cracks in concrete surfaces (Tomoyuki Yamaguchi , 2008). They proposed an image-percolation model that identifies cracks by referring to the connectivity of brightness and shape of the percolated region. They had assumed cracks were composed of thin interconnected textures. The validity of the proposed work was given by using ROC analysis utilizing experiments on actual concrete surface images. Figure 5 (a) show two samples of concrete surface with cracks and (b) shows the results of both images after processing using percolation method. Samples images show inherent variations in greyish shade of concrete along with cracks and dents. It is also observed that the pixel intensities of cracks are comparatively darker than the background and similar to the intensities of dents. Even with the presence of dents, the results images shows that the cracks are identified accurately.



(a) Samples images with crack

(b) Result images

Figure 5 Sample and results images of percolation method. (Tomoyuki Yamaguchi , 2008)

Noises like shading spots, dents, blemishes, and divots are present on the concrete surfaces which make crack detection difficult. To remove such noises and detect cracks, a method was presented by Yusuke Fujita et.al. (Yusuke Fujita , 2006). The method includes subtraction pre-processing where a smoothed image obtained by convolving the original image with a median filter is subtracted by the original image. It removed the variation caused by irregular illuminated conditions or shading. Other noises were removed by line filter based on the hessian matrix. Receiver operating characteristics analysis was applied to 50 real concrete images to evaluate the performance of the method. The sample concrete image with crack and result images shown in Figure 6 confirms that the method is effective for crack detection in a practical situation. The sample image multiple shadows, noises and variations in the background color. The result shows that the cracks are identified correctly and the algorithm is able to overcome the shadows and noises.

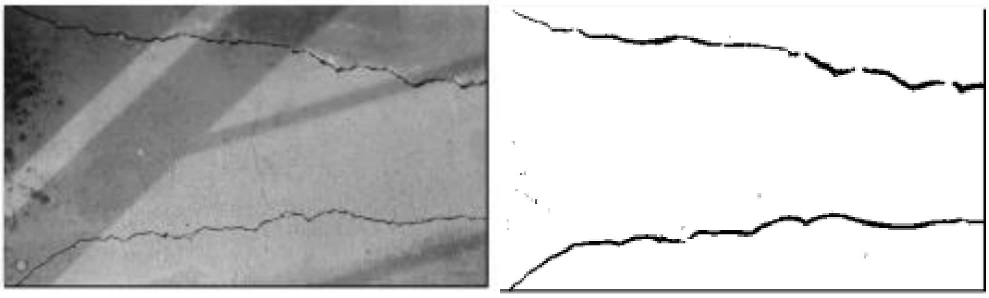
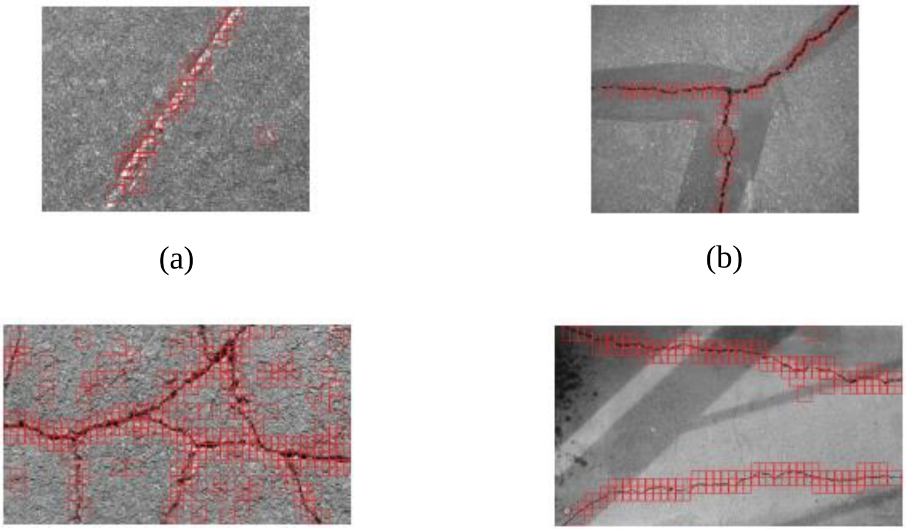


Figure 6 Sample and results images of line filter method. (Yusuke Fujita , 2006)

Ayaho Miyamoto et.al. (Ayaha Miyamoto , 2007) presented an approach for developing a system which can extract crack patterns from digital images using image processing. A crack sketching support system was discussed which includes steps like image acquisition, pre-processing, lens correction, measuring the crack length and crack width and producing a sketch of the crack pattern. A complete procedure to measure crack width from the difference in brightness was also discussed. Validation of the crack width measurement produced by the system was compared with manual methods. It was concluded that, with few allowable errors, the developed system can be able to produce satisfactory results.

Pingrang Wang et.al. summarized crack detection methods based on images and presented the comparison in four categories: integrated algorithm, morphological approach, percolation-based method, and practical technique. Experimental results were presented for each category of the method. It was concluded that the integrated algorithm is suitable for shading correction morphological approach gives better results compared to Otsu’s threshold, but it lacks the precision to detect cracked pixels (Pingrang Wang , 2010). Further, the percolation-based method gives the best results as it operated on the image locally but the processing time is more compared to other methods. The practical technique is a semi-automatic method which requires human intervention and can give excellent performance for different types of concrete images.

A general crack detection method which can work with images of different objects like metal workpieces, concrete structures, asphalt roads and can eliminate the effects of noises such as non-uniform illumination, shading stains, and natural texture was proposed by Han Hu et. al. (Han Hu , 2010). A Support Vector Machine (SVM) classifier was trained for crack detection on 1000 images with crack and 1000 samples of size 20 × 20 pixels grid for each class i.e., crack and non-crack. Hough transform was used to obtain the position and angle information of grid cells. ROC analysis presented by the authors shows that the proposed methods gave better results as compared to other features namely Principal Component Analysis (PCA), Border Greyscale Chain (BGC), Moment Feature (MF), Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP). Figure 7 shows the results of crack detection using Hough Transform based Feature (HTF) on four different surfaces. The authors have verified the method on images of different product, however few results images are shown. Figure 7 (a) is image of a metal workpiece, (b) and (c) are images of concrete surfaces while (d) is the image of asphalt road surface. The method is able to detect the cracks which are marked with read boxes despite of challenges like varying image resolutions, illuminated conditions and textures.



(a)

(b)

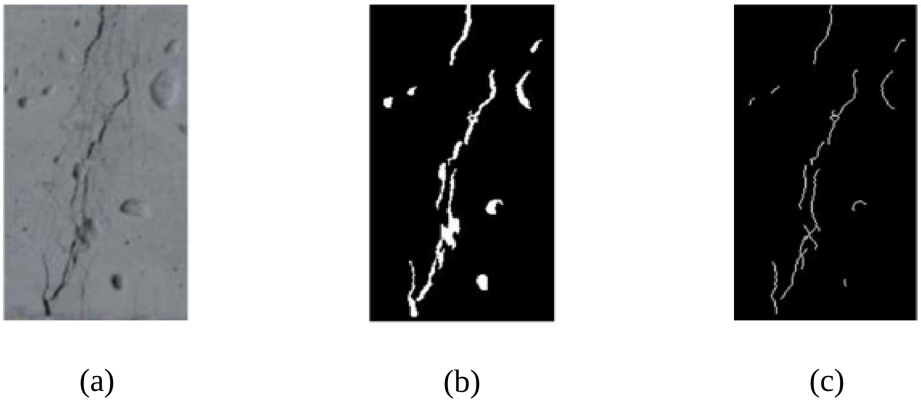
(c)

(d)

Figure 7 Crack detection results of HTF method. (Han Hu , 2010)

Hongke Xu et. al. presented their research work on an image segmentation algorithm for concrete bridge cracks (Hongke Xu , 2013). Histogram equalization and gaussian filter were applied after RGB to greyscale conversion to eliminate the effects of noise in the concrete crack images and enhance the images. Two image segmentation methods namely edge detection based on grey threshold iteration and edge detection based on canny iteration were implemented by the authors. The result analysis shows comparatively smaller errors for the Canny iterative method than for the greyscale threshold iterative method. Images of cracks at the bottom surface of a concrete bridge were used. The authors have reported experimental results in numerical form but no results images are found.

A method of crack detection using an otsu threshold and multiple filters were proposed by Ahmed Mahgoub Talab et. al. (Ahmed Mahgoub Talab , 2016). The proposed method is divided into three steps: the first being conversion from RGB to grey and edge detection and filtering using Sobel kernel, the second step was to binarize the image using a suitable threshold and filter out blobs with fewer pixels, and the third step was to apply otsu's threshold method to detect a crack. The images are captured during a uniaxial tensile test carried out on a laboratory concrete specimen. A digital camera was used to capture the images. The authors have not presented a performance evaluation of the proposed method and the result from images were shown. Figure 8 (a) shows the sample concrete surface image consisting of cracks, (b) shows the results image after processing while (c) shows the results image after processing and thining operation.



(a) (b) (c)

Figure 8 Result images of multiple filtering and threshold method. (Ahmed Mahgoub Talab , 2016)

Tran Hiep Dinh et. al. proposed a concrete crack detection method using computer vision (Tran Hiep Dinh , 2016). In the pre-processing, the authors implemented a line emphasis filter and histogram smoothing. The line emphasis filter uses Hessian matrix-based segmentation of objects into line-like and blob/sheet-like objects in the images. A histogram smoothing operation was performed using a moving averaging filter to remove random noise and assist in the automatic peak detection algorithm. Inspired by the practical problem of mountain exploration, the authors proposed automatic peak detection algorithms which were further divided into two steps: firstly search for observing location and secondly search for the highest peak in the closest distance. Figure 9 shows the sample image and crack detection results image where the cracks are overlayed in yellow color.

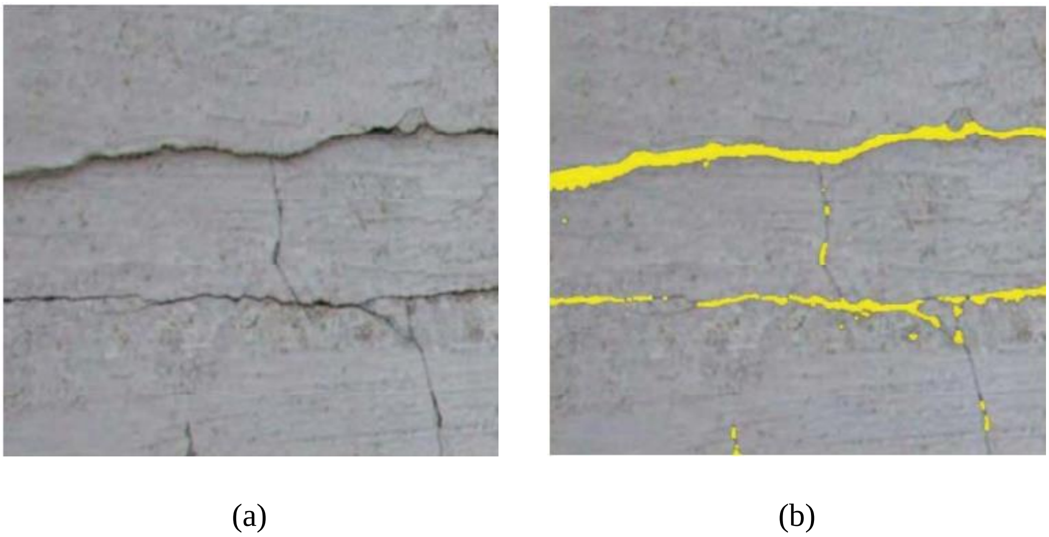


Figure 9 Result images of peak detection method. (Tran Hiep Dinh , 2016)

Algorithms implementation was presented in the work. The methods were tested on high contrast as well as low contrast images whose result images were shown in the paper. ROC analysis or performance evaluation along with a comparison with other known methods were missing in the paper.

Various authors have presented crack detection using a combination of different image processing-based methodologies. These methods cover thresholding, filters, segmentation, morphological approaches, edge detection and other derived or combined algorithms. The majority of the work was focused on a constrained surface image of concrete structures. These methods suffered a lot of limitations majorly due to inherent surface variations in concrete structures or members. Noisy and blurred images, dents, blemishes, and discolouration present on the concrete surfaces immensely affected the accuracy of algorithms. Due to this, AI, ML and other derived algorithms are proposed by researchers.

2.4 Artificial intelligence based approaches

Recently applications of AI-based systems have increased at a rapid rate. More and more problems are now being solved with the use of ML and DL-based approaches. Traditionally, applications which required classification or clustering for the data or information which has variations were solved by the AI approach. Now with the easy and cheaper availability of good computer resources, Amazon Web Service (AWS), Google Colab and others have increased the popularity of DL-based applications in industrial applications as well as among researchers and academicians around the world.

Concrete crack detection is a similar trivial problem which has quite a few variations in the data/information. Variations namely different shades of grey due to cement base colour, different colour wall paints, shadows, noise on the concrete surface, non-uniform light conditions, and blur poses a strong challenge for automatic crack detection and structural health monitoring using the power of computer vision as a whole. A solution approach which is nowadays widely explored by researchers around the globe is the usage of DL-based CNN models for crack detection. Strong contribution in the area of crack detection using ML/ DL/ANN is discussed below in detail. CNNs are a subset of DL which refers to the collection of algorithms and network which forms a model for large data set classification or clustering. CNN are popular with image classification and clustering applications as it involves 2D convolution layers.

These layers reduce image data and remove the need for manual feature extraction from images making them more suitable for computer vision applications. The only disadvantage the DL approach has over ML methods is the requirement of a large amount of labelled data, for example, image classification applications popular models like VGG, RESENT, Mobilenet, and Inception use millions of images to train the CNN model. ML based image classification applications require manual extraction of important and relevant features, which becomes a tedious and time-consuming job. However, in contrast to ML, DL performs “end-to-end learning” – where a network is given raw data and a task to perform, such as classification, and it learns how to do this automatically.

More problems are now being solved with the use of ML and DL based approaches. Yuanfeng Duan et. al. presented their work on damage detection in tied-arch bridges using a CNN (Yuanfeng Duan , 2019). The authors used Fourier Amplitude Spectra (FAS) of ambient wind vibration data combined with CNN. The detailed classification of different methods which have been proposed by authors to inspect or detect defects and or cracks in concrete structures or elements is shown in Figure 10 (Christian Koch , 2015).

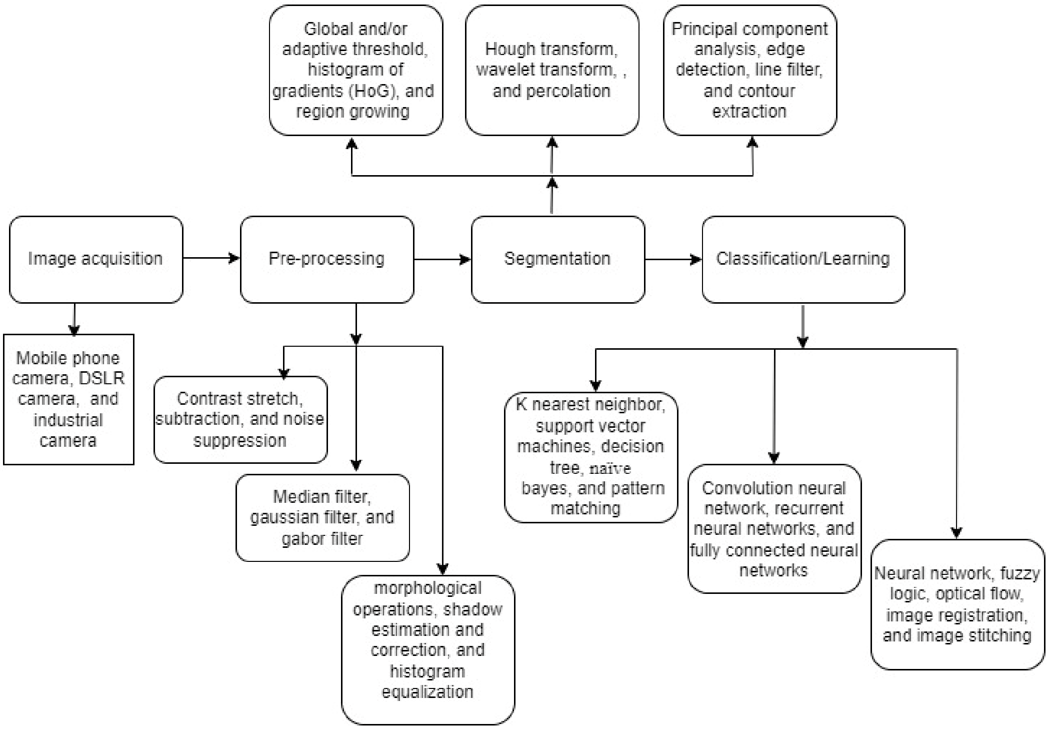

Global and/or adaptive threshold, histogram of gradients (HoG), and region growing
Hough transform, wavelet transform, , and percolation
Principal component analysis, edge detection, line filter, and contour extraction
Image acquisition
Pre-processing
Segmentation
Classification/Learning
Mobile phone camera, DSLR camera, and industrial camera
Contrast stretch, subtraction, and noise suppression
Median filter, gaussian filter, and gabor filter
morphological operations, shadow estimation and correction, and histogram equalization
K nearest neighbor, support vector machines, decision tree, naïve bayes, and pattern matching
Convolution neural network, recurrent neural networks, and fully connected neural networks
Neural network, fuzzy logic, optical flow, image registration, and image stitching

Figure 10. Classification of vision-based methods to detect cracks in concrete.

Discrete wavelet transform based damage classification of concrete structures namely concrete bridge and laboratory elements were proposed by Shahid Kabir et. al. (Shahid Kabir , 2007). Cracking, steel corrosion, spalling and erosion were the damages under focus. Haar’s discrete wavelet transform combined with a grey-level co-occurrence matrix was deployed for the classification. Maximum Likelihood Classifier (MLC) and k-means clustering were the two separate approaches used for the classification and quantification of the damage. The MLC method provided accuracy ranging from 76% to 79% and proved much more accurate as compared to k-means clustering.

DL-based crack damage detection using CNN was proposed by Young-Jin Cha et. al. (Young-Jin Cha , 2017). The authors presented work on crack detection which is less influenced by noises caused by varying light conditions, shadow, casting, and blur. A CNN was trained on 40,000 images of 256 × 256 size. MatConvNet, a CNN developed on MATLAB was used to perform the study. The images were taken from a complex building at the University of Manitoba using a hand-held Nikon D5200 DSLR camera. The authors used the sliding window method to scan larger images. The overall architecture of the network along with the description of each layer was well presented in the work. An accuracy of 97% was claimed by the author.

Automatic crack measurements were carried out by Tung-Ching Su et.al for the building surfaces and concrete pavements using the morphological segmentation-based edge detection-II method (Tung-Ching Su , 2018). The primary aim of the work was an enhancement of image segmentation for accurate crack segmentation and measurement of the length, width and area of cracks. The proposed method was cross-compared with and without Cross Curvature Evaluation (CCE). Very minor errors were reported in the measurement of crack length, width and area. The study aimed at assisting the experts in the surface inspection of concrete structures.

NB-CNN: DL-based crack detection using a CNN and Naïve Bayes data fusion was presented by Fu-Chen Chen et. al. (Fu-Chen Chen , 2018). It analyses video frames of nuclear power plant components for crack detection. A CNN is proposed to detect crack patches in each video frame. Naïve Bayes was used to discard false positives and the data fusion technique was used to maintain the spatiotemporal coherence of cracks in videos. Videos of 20 underwater components of the nuclear power plant were captured and a dataset of 5326 images of 120 × 120 was extracted and manually annotated. Orientation and brightness variations were manually generated. Patch-based and frame-based performance evaluation was presented. The proposed method achieves 99.9 % True Positive Rate (TPR) against 0.1% False Positive Rate (FPR). The proposed method was compared with LBP-SVM and other standard methods.

The CNN approach for robust structural damage detection and localization was presented by Nur Sila Gulgec et. al. (Nur Sila Gulgec , 2019). The CNN topology was designed to classify simulated damaged and healthy cases and localize the damage when it exists. The performance of the proposed technique was evaluated through the finite-element simulations of undamaged and damaged structural connections. Samples were trained by using strain distributions as a consequence of various loads with several different crack scenarios. Completely new damage setups were introduced to the model during the testing process. Based on the findings of the proposed study, the damage diagnosis and localization were achieved with high accuracy, robustness, and computational efficiency.

Patch-based crack detection using a CNN was proposed by Somin Park et. al. (Somin Park , 2019). The system used a car black box camera to capture images of the road and detect cracks on the road. The acquired images were trained for road extraction and crack detection. The proposed framework first extracts the road surface from the acquired image, the extracted portion is cropped and passed through the crack detection network to detect cracks on the road surface. The developed CNN achieves an accuracy of 98.33% in road extraction along with 97.74% precision and 95.21% recall with 120 test images. A precision of 90.81%, 74.4% recall and 81.79% F1 score was achieved for crack detection.

An improved CrackNet based automated pavement crack detection on 3D asphalt surfaces was proposed by Allen Zhang et. al. (Allen Zhang , 2018). The proposed modified CrackNet architecture show improved learning capability and provides faster performance. The images were collected using a Digital Highway Data Vehicle (DHDV) developed by WayLink. The vehicle has 3D sensors which used the 3D laser triangulation principle to capture the 3D profile of the road surface with a resolution of 1mm. More than 6000 3D pavement images were collected for the training, testing and validation of the proposed CrackNet II model. Testing carried out on 200 images shows that CrackNet II outperforms CrackNet architecture. A precision of 90.20%, recall of 89.06% and F1 score of 89.62 were obtained with the testing. These parameters show that the CrackNet II provides better performance and can detect finer cracks.

CNN based robust pixel-level crack detection was proposed by Mohamad Alipour et. al. (Mohamad Alipour , 2019). CrackPix, was developed by the authors to detect cracks at the pixel level. Images samples were collected from bridges, buildings, and sidewalks. Manual labeling at pixel level on the captured images was carried out

MATLAB image labeler tool. CrackPix was trained on 90% of image samples of the labeled data. Testing carried out on 10% of the image dataset showed a precision of 91.24%, recall of 92.17% and 91.70 F1 score which was compared with VGG16. The comparison showed that the CrackPix performs better and the network output provides the measurement of crack width and length. The authors have presented the entire model architecture, collected sample images, results, performance evaluation and comparison in depth.

A multi-classifier was proposed by Philipp Hüthwohl et.al. for the classification of defects encountered on reinforced concrete bridges (Philipp Hüthwohl , 2019). With the objective of automatic inspection of reinforced concrete bridges, the authors proposed a three-stage, multi-classifier which can classify potential defected bridge areas in conformity with existing bridge inspection guidelines. A dataset containing manually acquired bridge images and an inspection database of the transportation department were collected for the training and testing of the classifier. Three separate pre-trained deep neural networks were fine-tuned and re-trained using the collected image database. The authors claim that the method can consistently classify the multiple bridge defects with an average mean score of 85%.

A robust concrete crack detection method which uses DL was proposed by Donghan Lee et. al. (Donghan Lee , 2019). The authors proposed a crack image generation algorithm based on a 2D gaussian kernel to overcome the lack of crack image data unavailability. Data augmentation techniques like random flipping, random cropping, and random hue and contrast manipulation on the flipped image were used by the authors. A patch-based CNN was trained on roughly 27,000 patches of crack and non-crack parts. The proposed method achieved an accuracy of 97.98% with the actual dataset, and 98.18% with the actual and simulated dataset. The authors have presented results of various crack images along with variations in contrast and hue along with a comparison of computation times.

CrackNet-V, a DL-based approach was proposed by Yue Fei et. al. (Yue Fei , 2019) for pixel-level cracking detection on 3D asphalt pavement images. CrackNet-V is inspired by CrackNet and VGG and has a deeper architecture with fewer parameters. The network was tested on 500 testing images and provided 84.31% precision, 90.12% recall and 87.12 F1 score. A digital highway data vehicle was used to collect the 3D data sample of pavements. The vehicle was mounted with a PaveVision3D system developed by WayLink, to acquire full lane scale 3D data of the highway with a resolution of 1mm. The authors have presented the network architecture, 3D images of the lane, a comparison of CrackNet and CrackNet-V, training and validation performance and result images. The authors have presented a new activation unit called "Leaky Rectified Tanh" which is used in place of "Leaky ReLU". The overall performance of CrackNet-V is better specifically in detecting fine cracks as compared to CrackNet.

Suguru Yokoyama et. al. (Suguru Yokoyama , 2017) presented their work on a ML based automatic concrete crack detector. A CNN was developed to classify concrete images into five categories namely: crack part, chalk letter part, joint part, surface part and other parts. The dataset was developed manually from 2000 images of cracked concrete. Horizontal inversion and image resizing were used to convert the dataset images into a size of 64 × 64 pixels. The author claims that the developed classifier can classify the cracked part in concrete with no stains whereas it fails to accurately detect in the case of concrete with stains. In a few of the cases, results depicted that, stains were detected as cracks. The author further suggested including images of concrete with stains to improve the classification and perform pre-processing operations.

It can be concluded from the crack detection work discussed above that the performance of CNN-based approaches is much better as compared to conventional image processing-based algorithms. However, a point of similarity was found between both types of approaches. Both the type of approaches built their custom dataset or worked on a specific area and type of structures or concrete elements. The custom build dataset met the specific requirements of the proposed crack detection algorithms. Due to the variety of large infrastructure surfaces worked upon for crack detection, no generalized crack detection algorithm was observed so far. Different types of image acquisition devices were used, but industrial standard devices and the use of specific illumination conditions were not observed in the literature work. The research work in the area of crack detection using AI can be extended to include assessment criteria to assess the performance such as the computation time, computation resource, advanced hardware and applicability in real-time scenarios. Limited work was observed to measure crack properties, the severity of damage and prediction of the serviceability.

2.5 Crack properties measurement

Defects in concrete structures can occur during and after construction due to various reasons. These defects are primarily observed as surface cracks and evaluation of these cracks can assist in predicting the need to repair, reinforcement or other measures (Stephanie German , 2012) (Zhenhua Zhu , 2011). However, the primary task in the area of structural health monitoring would always be the accurate and reliable detection of cracks. On the other hand, detecting cracks alone may not provide a fruitful conclusion related to extent of damage and possible measures required. From the literature presented, it can be concluded that using DL-based methods, CNN accurate detection of cracks is achieved (Jung Jin Kim , 2020). The researchers have started to shift the focus towards the crack properties measurement and assess the extent of the damage (M. M. Sarker , 2017). The crack properties measurement should be carried out both in real-life concrete structures as well as in laboratory-scale concrete elements. Crack properties like the location of cracks, the orientation of cracks and type of crack, and structural or non-structural cracks are important parameters for the total assessment of the health of a structure.

Width, length, depth, and area are the conventional crack properties which are of prime importance looking into structural integrity prospects. But cracks are never straight and their area varies constantly. Hence, measuring the length and area of a crack is a very cumbersome, time-consuming process. The measurement will be highly inaccurate. Similar to a crack length, width and depth are also inconsistent throughout the length of a crack. Point-by-point measurement of crack width and crack depth is carried out which is a manual and slow process. The accuracy of results not only depends on the specifications of the instrument used but also depends on the skills of the engineer.

A better and more detailed understanding of concrete structure's behaviour turns out to be the next step after accurate and robust crack detection on such surfaces. To understand the behaviour of various concrete structures under loading, measurement of crack properties becomes extremely important. Crack properties like the number of cracks in a given surface area, crack length, crack width, crack area, crack depth, location of cracks, and the orientation of cracks are vital properties which depict the extent of damage and shows the performance of concrete structures. Physically measurable properties like load, force, strain, displacement and others are possible up to a certain extent using state-of-the-art sensors and sophisticated data acquisition systems. Such physical properties are measured for the concrete and not for the cracks and hence do not provide information on crack properties. Modern building, steel structures, and high-rise building deploys building automation and focus on the physical measurement of properties like a strain at critical locations (Yong Xia , 2014) (Hyo Seon Park , 2013).

Measurement of concrete crack properties using vision is difficult in the case of practical concrete structures like bridges, buildings, and pavements due to their large structures and large field of view for a good quality camera. But the same is certainly possible at critical locations having limited portions and a relatively smaller field of view. In the case of laboratory concrete structures or comparatively smaller concrete elements like nuclear storage vessels, vision-based measurement or estimation of crack properties is more likely possible. It will provide a better understanding and prediction of concrete structures and elements.

A method for measurement of crack length and width in concrete construction was proposed by Cao Meilli et. al. (Cao Meili , 2013). Averaging and binarization were used to pre-process the images. Methods to calculate crack length and width were presented along with testing and field experiments. No mention of the truth value, number of images considered to validate the methods is limited.

Liang-Chein et al. (Liang-Chien , 2006) presented their work on the measurement of crack length and width extracted from multitemporal images of concrete surfaces. The proposed method of measurement needs a user to select seed points to indicate cracks on the images. The author claims that the system improves the degree of automatic extraction and recording of the length and width of cracks.

Numerical representation of defects was carried out by R S Adhikari et. al. (R. S. Adhikari , 2014). The authors presented an integral model to quantify cracks, detect changes and used neural networks along with a 3D visualization technique to visualize the defects. The crack length was evaluated using the perimeter of the skeleton of the crack whereas the change detection model was based on the Fourier transform of the digital images. Crack depth is predicted by the neural network model considering the crack length and change detection. The authors concluded that the developed models can be integrated into a Bridge Management System (BMS) to enhance the reliability of the decision-making process.

In order to mitigate the issues associated with the failure criteria of asphalt pavements, Barzin Mobasher et. al. presented their work on the evaluation of crack propagation properties applied for asphalt mixtures (Barzin Mobasher , 1997). The compliance approach and R-Curve approach were used to evaluate the low-temperature fracture parameters of conventional asphalt concrete and asphalt rubber mixture. Multiple beam specimens were prepared with different binder contents and tested under three-point bending flexural conditions at two test temperatures. The authors collected nonlinear fracture parameters by testing the specimens using a closed-loop servo-hydraulic system with a crack mouth opening as the control parameter. It was concluded that the asphalt rubber mixture has higher fracture toughness, improved resistance to cracking, and is less sensitive to temperature as compared to the asphalt concrete.

Khalili et. al. introduced new methods to analyze and measure the crack dimensions in solid materials (Khalili , 2015). The authors presented two methods to measure the crack dimensions in mechanical tools and bricks using machine vision. One of the methods used an image skeleton to measure the length while the second one used the perimeter of the crack which is halved to obtain the crack length. The authors proposed a new connectivity algorithm to overcome the crack length measurement error which was 2.6% and 2.0% when 4 connectivity and 8 connectivity were used respectively. The new connectivity algorithms showed a 1.3% error in the measurement of crack length. The authors also used a skeleton based method to measure crack length. The authors concluded that the measurement of crack length via the perimeter method is less time-consuming while the skeleton method has higher accuracy.

A mobile robot based crack detection and crack measurement system was proposed by Seung-Nam Yu et. al. (Seung-Nam Yu , 2007). The crack detection was performed in a tunnel by using a CCD camera and using image processing to detect and measure cracks. The authors proposed an image processing routine to extract geometric properties and patterns of the crack to ensure accurate crack recognition.

In order to address the inaccuracies associated with crack discontinuities, Shuai Zhao et. al. proposed a DL model for crack segmentation (Shuai Zhao , 2021).The authors worked on shield tunnel lining images and used PANet DL model to segment cracks. The authors also showed the computation of width and shorted crack length from segmented binary images using the A* algorithm. The authors have shown a comparison of the proposed method with Mask R-CNN, U-Net, and DeepCrack where the proposed method showed superior performance in terms of overcoming the crack disjoint problem and skeletonization errors.

Nicola Gehri et. al. proposed (Nicola Gehri , 2022) a method for crack behaviour in large-scale experiments with complex crack patterns with the use of Automatic Crack Detection and Measurement procedure (ACDM) based on surface displacement measurements obtained with Digital Image Correlation (DIC). The authors proposed refinements in crack detection and crack kinematic measurement. The proposed method is validated on large scale 2.0 × 2.0 meters shear panel experiments consisting of highly complex crack patterns. Crack properties like width, slip, inclination and spacing in large-scale concrete elements tests were extracted to study the behaviour of the elements. The authors claim that the proposed method shows improved performance in terms of fine crack detection and detect the locations of crack with higher accuracy.

In addition to the literature discussed here, other works in the field of crack properties measurement have been carried out by researchers. A pixel-wise crack segmentation methodology was proposed by Yiquing Liu et. al. wherein the authors utilized a CNN and Active Contour Model (ACM) to perform crack segmentation (Yiqing Liu , 2021). Chaobo Zhang et. al. proposed a framework to retrieve accurate geometric properties of each crack segment by recognizing the crack's inherent branching patterns (Chaobo Zhang , 2022). The authors used an Unmanned Aerial Vehicle (UAV) to perform a field test on a cracked concrete wall.

Measurement of crack geometric properties and identification of their correlation was addressed by researchers. Image processing based are widely popular in the domain and the use of AI/ML based methods has been increasing. A lot of other techniques like impact echo technique, radar cross-section dip and ultrasound diffusion were used to measure the depth of cracks. However, these methods suffer some limitations like the cost of setup, manual procedure and complex information retrieval algorithms. The measurement of properties depends upon the methodology used for the measurement and the surface under observation. However, advanced techniques have the potential to overcome the limitation of presently available methods at the same time would provide online results.

2.6 Indian knowledge system

Crack detection and its associated works have been carried out in India by various researchers. Although major contributions and work in the area of crack detection had started from Japanese, Chinese, American, Canadian and Korean authors. The primary reason for this fact may be due to the effects of natural calamities in the country, the severity of earthquakes, rapid infrastructures being built, collapse and other issues affecting human life. Primary contributions from Indian authors in the area of crack detection with respect to the Indian perspective are discussed here.

Milind Padalkar et. al. carried out substantial work in the heritage monuments by identifying defaced regions, and cracks and performing auto-inpainting. The authors have contributed to the area of digital heritage reconstruction starting from 2015. The work is focused on the automatic detection of cracks, defaced regions, and vandalized regions in facial images of statues and heritage monuments. Methods like singular value decomposition, bilateral symmetry and textons features based method and Scale Invariant Feature Transform (SIFT) and homography based method were proposed in their works (Milind G, Joshi and Khatri , 2017) (Padalkar and Carlos Beltrán-González , 2021) (Milind G, Zaveri and Joshi , 2012).

Milind Padalkar et. al. also presented a framework for the automatic detection and infilling of cracks in images and videos of heritage monuments. The authors discussed the need to digitally repair photographs of damaged or vandalized heritage monuments. The authors used a measure derived from edit distance and compared image patches to detect the cracks in digital images. The proposed method was extended to perform inpainting in videos by using SIFT and homography. The authors used videos from a heritage site in Hampi, India (Padalkar and Joshi , 2015).

An overview of different conventional and non-conventional methods of structural health monitoring was discussed by Arabinda Sharma et.al. (Arabinda Sharma , 2016). A variety of other non-destructive evaluation methods based on image processing was discussed along with useful outcomes, constraints, and challenges. Applications of image processing to monitor structure health has matured and offers non-contact measurement, good amount area, mm accuracy in large structures and micron accuracy in laboratory scale structures.

A similar comprehensive review of crack detection methods was presented by Arun Mohan et. al. (Arun Mohan , 2017). The authors have covered a brief review of the various types of works presented by researchers in the area of crack detection. The authors have presented a review of 50 research papers related to crack detection and covered detailed analysis including objectives, accuracy level, error level, and the image data sets. The authors highlighted the research gaps and issues associated with crack detection based on the analysis.

Fuzzy logic and a neural network-based model were applied by Gajanan K. Choudhary et.al. to detect cracks in concrete (Gajanan K Choudhary , 2012). Using edge detection, features were extracted from the images of concrete surfaces which were fed into models for detecting cracks. Models were developed to classify objects into cracks and noise and were tested on 205 images. Area and major-minor axes ratio properties of objects were fed into fuzzy logic to classify objects into cracks and noise. It was concluded that the neural network model gives the most accurate results as compared to fuzzy models. Performance measures of model accuracy were used in the validation. Details on the training time of models as well as computation time were missing. No details on the training time of models as well as computation time were provided. Also, details on the practical implementation of the models are missing.

Suresh bhalla et.al. (Suresh Bhalla , 2011) presented a defect detection using a thermal imaging technique study on two concrete specimens-beams and slabs. The results of thermal imaging were compared with that of the ultrasonic pulse velocity technique. Defects as small as 50 mm in width were detected and located by thermal imaging.

A semi-automatic crack length and width measurement method was presented by T Barkavi et. al. (T. Barkavi , 2019). The authors presented a new image processing algorithm to detect cracks and measure the length and width of the crack which is implemented in MATLAB. The algorithm flowchart, pseudocode, results, validation of the algorithm, comparison with the method presented in literature and uncertainties handled by the algorithms were discussed in the work. The authors reported an average error of 1.02% in the measurement of crack length, and a 2.6% average error in the measurement of crack width. The crack length and width measurements with the use of an algorithm were compared with the manual measurement.

Diana Andrushia et. al. proposed an auto-associative neural network to detect cracks in concrete structures (Diana Andrushia , 2020). Diana Andrushia et. al. also proposed an autonomous crack detection framework for concrete structures post-fire conditions in their work (Andrushia, Anand and Neebha , 2022). The authors have presented the study using multiple samples of high strength Self-Compacting Concrete (SCC) which were exposed to fire in order to induce thermal cracks. The concrete cube samples were heated in a digital computer controller electric furnace following the relevant ISO standards. A hybrid network comprising a CNN and a Long Short Term Memory (LSTM) network was used to detect the damage. The authors gave details of the framework, network architecture, dataset, model training, validation and testing. The accuracy, precision and recall reported for the proposed method were compared with other known methods wherein the proposed method showed better results.

The authors have also presented similar work on thermal crack detection on a concrete structure which was exposed to elevated temperature using DL (Diana and Lubloy, Deep learning based thermal crack detection on structural concrete exposed to elevated temperature , 2021). The authors Diana Andrushia et. al. also presented work on a novel approach to the use of ripplet transform applied to detect and quantify thermal cracks in concrete structures (Andrushia and N. Anand , 2020). Research work on image denoising and segmentation was also presented by Diana Adhrushia et. al. (Diana, Anand and Arulraj , 2019). The use of numerical correlation analysis was proposed to detect thermal surface cracks in structural concrete by Andrushia A. Diana et. al. (A. A. Diana, N. Anand and R. Walls , 2022). The knowledge of deep CNN and transfer learning was applied by Diana Andhrushia et. al. for shrinkage crack detection in expansive soil (Andrushia, Neebha and Umadevi , 2022).

The research work in the area of crack detection is presented by Indian authors covering the use of different algorithms and methods including CNN-based approaches. Few authors have also presented a detailed literature review covering the different contributions in the area of crack detection based on images. Crack detection-related work was carried out on statues in historic monuments, expansive soil, fire-exposed concrete cubes, and others. From the review presented, it is observed that crack detection on roads, pavements, tunnels, nuclear vessels, bridges and other surfaces which are well-explored in developed countries is not targeted in India up to a certain extent. The gaps can be addressed in the future by researchers from India considering the issues faced in new and old infrastructure in India. In addition to that, crack detection in concrete cubes performed online during a compression test is not explored by the Indian research community. It justifies the research work presented in this thesis. Also, work related to crack monitoring and analysis with respect to applied load is not observed in the literature discussed above.

2.7 Summary

From the literature review, it is observed that significant work has been carried out to detect a crack in different structures, measure crack properties and assess the damage in infrastructures. Most of the researchers have applied and tested the algorithms either on offline image sets or in a constrained environment to validate the algorithms. But in practical implementation, the image processing algorithms will be affected by lighting conditions, changes in the setup environment and a lot of other factors. Also, computation time in image processing algorithms is one of the important performance factors, especially in the case of online implementation. A good portion of the presented work is carried out in an offline situation, and under a constrained environment with assumptions. Very limited work was found with real-time implementation and with a usable setup. Researchers have rarely explored the possibilities to characterize cracks with respect to load. Although many authors have presented their work on crack detection and defect detection on various surfaces including concrete, asphalt, and nuclear vessels, limited contribution was observed specifically for the laboratory scale concrete elements for example a concrete cube which is widely used in a compression test.

The concept of DL can be well deployed for image-based applications where the object under observation has unmeasurable variations. The conventional image processing-based method presented by the authors suffered the limitation of variations in light conditions, random noise on the concrete surface, shadow, blur and other factors. The limited literature on real-time crack detection, crack length, width, and depth measurement proved the need to develop a sophisticated system addressing the challenge. The results of the work will be useful not only in terms of research of improved crack detection and its properties measurement algorithms but also facilitate automated testing of concrete elements. Crack detection and its associated measurements will be extremely useful in understanding concrete behaviour under different kinds of loads. Further, the outcomes can be useful to detect cracks accurately in many practical situations like dams, bridges, tunnels and various other concrete structures.

Looking at the limitation of the research work proposed in the area of concrete crack detection and structural health monitoring, CNN-based crack detection in laboratory-scale concrete cubes is implemented. The crack detection is synchronously carried out with respect to the applied compression load while the concrete cube is subject to a conventional compression test. Along with CNN based crack detection, measurement of crack properties such as the number of cracks, length, width, area, and location is performed on concrete cubes of different compressive strengths. The properties were measured with respect to the applied compressive load. The data is monitored throughout a compression test and analyzed for fruitful observations. A CNN based crack detection may not be a novel approach, but when the methodology is applied for crack detection in laboratory-scale concrete cube images, it becomes a different application of the methodology. Monitoring crack growth and propagation with respect to applied load is another innovative aspect of the work. Finally, analysis of the crack properties with respect to load carried out for concrete cubes of different compressive strengths can provide vital observations related to the performance of the concrete cube.

Chapter 3 Experimental setup

Chapter 3 of the thesis focuses on the development of an experimental setup in order to develop a concrete crack detection system. The detailed requirements are presented in the chapter along with the hardware and software essential for the development. The technical specification of the important hardware is covered with appropriate images of the parts along with necessary data. Software platforms and libraries used for the development are also covered in detail in addition to the details on the software requirements. The chapter also covers the integration aspects of the overall system in order to meet the requirements. The chapter ended with the images of the experimental setup, images of concrete cubes, and load data acquired using the developed experimental setup.

3.1 Requirement study

The crack detection can be performed on different types, shapes and sizes of concrete and other surfaces of infrastructures prone to damage. Various methods like physical and visual inspection, acoustic-based inspection, vibration-based method, ultrasonic pulse velocity-based method, and other electric and magnetic methods (M. J. Anitha , 2021). The primary aim of crack detection using images is to automatically detect cracks from the surface images of concrete surfaces under observation.

With the reference to the objectives and scope of work stated in chapter 1, the present work focuses on crack detection in surface images of standard concrete cubes subjected to compression testing in the laboratory environment. Figure 11 shows a complete block diagram of the proposed computer vision system implemented at the Heavy Structures Laboratory of Nirma University for compression testing of concrete cubes.

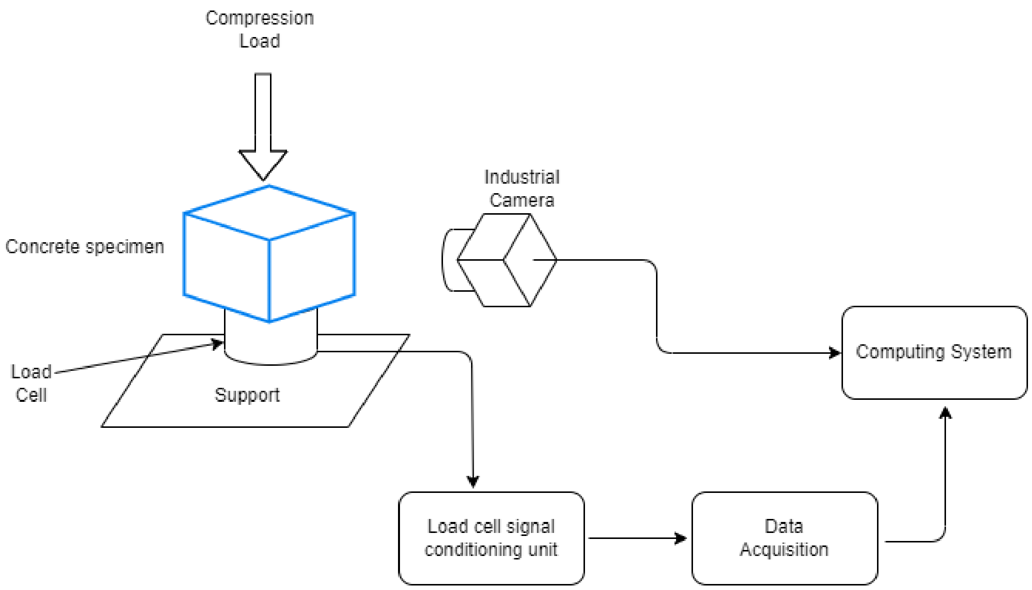


Figure 11. Crack detection methodology block diagram.

The primary step in the work is to develop an experimental setup which can perform the initial task of data acquisition. The experimental setup should be designed to acquire synchronous data of image and applied load during the compressing testing cycle of a concrete cube. The detailed requirements of the experimental setup are mentioned below.

1. A compression testing machine
2. Standard concrete cubes for testing
3. Apparatus to acquire image data
4. Uniform illumination over the observed surface area
5. Instrumentation to acquire compression load data
6. Mounting assembly to hold the apparatus and instrumentation
7. Electric power supply, communication cables and necessary field wiring
8. Computing resources to integrate and synchronize the hardware
9. Software and libraries to facilitate synchronous data acquisition, processing, visualization and data logging.

Looking into the requirements of the experimental setup stated above, the detailed hardware and software are selected. These components are then integrated with the software package to fulfil the requirement and achieve the objective of the present work.

3.2 Hardware selection

The development of an experimental setup plays a major role in the implementation of the overall system. The hardware requirement of the setup was divided into three major categories comprising a means to generate cracks, means to capture the real-time data and resources to process the data. A standard compression testing machine is a widely used instrument in civil engineering laboratories. The testing machine is used to carry out a compression test on different concrete elements and measure the compressive strength of the element under test.

In the present work, standard concrete cubes of size 150 mm × 150 mm × 150 mm are used. A 3000 kilo Newton compression testing machine which is load controlled is used in the present work. The testing machine comprises a loading unit, pumping unit, controlling and indicating unit arranged together with upper, lower and side plates. The controlling unit can automatically operate the pump, control the set loading rate and switch off the machine under predetermined conditions. The controlling and display unit shows the compressive strength of the concrete cube under testing and also allows the user to configure parameters like loading rate, sample data and others.

Various standard codes recommend a concrete cylinder or concrete cube as the standard specimen for the test. For the cube test, two types of specimens either cubes of 150 mm × 150 mm × 150 mm or 100 mm × 100 mm × 100 mm depending upon the size of the aggregate are used. For most of the works cubical moulds of size, 150 mm x 150 mm x 150 mm are commonly used and hence, the present work uses the same size of concrete cubes. The concrete cube sample is prepared using the cube moulds and allowed to cure for 28 days. The samples were prepared and tested according to the Indian standards IS:516. According to IS 516, the load should be applied gradually at the rate of 140 kg/cm^2 per minute till the specimens fails. The load shall be applied slowly without shock and increased continuously until the resistance of the concrete cube to increasing load breaks. Compressive strength can be obtained by calculating the load at the failure divided by the area of the specimen. According to a recent update in IS 516 Part 1 Section 1 2021, the rate of loading is revised as 14 N/mm^2 per minute.

However, for a concrete cube of surface 150 mm × 150 mm × 150 mm, the loading rate can be 5.25 kN/s.

Image acquisition is an important process in an image processing-based application. The acquisition should be carried out similarly as in the case of deployment to maintain consistency in both acquisition and processing. The majority of the research work reported the use of portable and commercial image acquisition devices like DSLRs, mobile phones, and other commercial cameras. These devices are often operated in an auto mode wherein the image acquired in different conditions will not be consistent. And hence, the development of image processing algorithms will be even more challenging. Moreover, images acquired with such devices may have shadows, non-uniform light, blurriness, and noise. Additionally, the surface under observation may contain dents, spots, noise, discolouration, blemishes, and rust marks. Thus, in the present work, an industrial camera coupled with fixed focus is utilized along with white lights.

The use of an industrial camera will produce consistent images under the lights in order to overcome the challenges and issues mentioned above. The compression testing of the concrete cube samples will be carried out in a laboratory environment. Hence, the use of an industrial camera is appropriate in these conditions. Such cameras are also called machine vision cameras as they are predominantly used in the industrial inspection system. As the concrete cube surface observed by the camera is steady, a rolling shutter camera is used. Any one surface of the concrete cube is observed by the camera. The size of the concrete cube front surface which is monitored by the camera is 150 mm × 150 mm. The Field Of View (FOV) is considered as 180 mm × 180 mm as the cube will move slowly in the vertical direction as the load progress during the compression testing. The additional tolerance of 10 mm on each side is considered so that the edges of the concrete cube can be identified easily. However, as the camera sensor is rectangular, the actual field of view is 239 mm × 180 mm, the working distance is 215 mm and the focal length required is 5 mm. The additional view in the horizontal direction can be cropped by defining Region Of Interest (ROI) X and Y as per requirement.

The camera, lens and lights specifications are mentioned in Table 1 (daA2500-14uc (CS-Mount) - Basler dart , 2022) (Basler Cable USB 3.0, Micro B sl/A, P, 3 m - Data

Cable , 2022) (MPZ Machine Vision Series H0514-MP2 , 2022) (Syska Slim LED 15W RDL Square Downlight-6500K , 2022). A 5 Mega Pixel (MP) resolution, Complementary Metal Oxide Semiconductor (CMOS), area scan, colour and USB3.0 camera is chosen for the vision system.

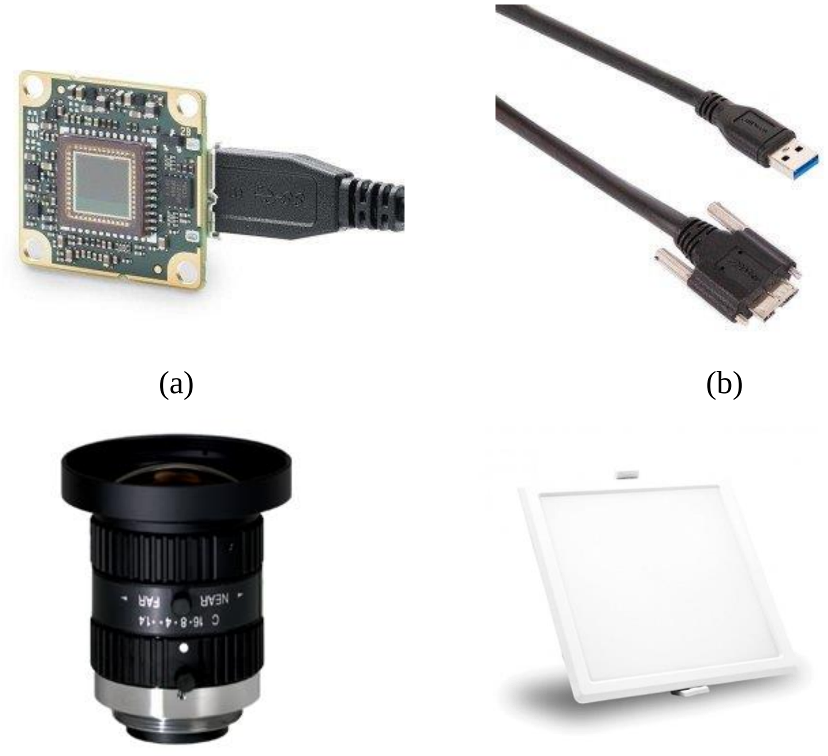
Table 1. Vision hardware specifications.

Industrial camera	**Lens**	**Lights**
Model and Make: dA2500-14uc, Basler, Germany	Model and Make: H0514-MP2, CBC, Computar Japan	Model and Make: SSK-RDL-S-15W-6500K, Syska, India
Resolution: 5 mega pixels	Sensor format: $\frac{1}{2''}$	Led
Sensor format: $\frac{1}{2.5''}$	Focus: Fixed focus	Color: White
Communication: USB3.0	Focal Length: 5 mm	Operating voltage: 220 volts AC supply, 50 Hz
Shutter type: Rolling shutter	Iris: Manual iris with screw lock	Dimension: 165 mm × 165 mm × 35 mm
Pixel size: 2.2 um × 2.2 um	Focus: Manual focus with screw lock	Diffuser: Yes (medium)
Sensor size: 5.7 mm × 4.3	C- mount	Power: 15 watts
Image resolution: 2592 pixels × 1944 pixels	Angle of view: D – 76.7°, H – 65.5°, V – 51.4°	Current: ~200 mA
Frame rate: 14 fps	Aperture ratio: 1-1.4	Luminance: 1500 lux
Pixel bit depth: 8,12 bits	Distortion: 0.48%	----
Sensor: MT9P031, ON Semiconductor	Operating temperature: -10 °C to +50 °C	----
Synchronization: hardware trigger, free run, software trigger	Weight: 102 grams	----
Programmable gain, exposure time and other parameters	----	----
Mono/Color: Color	----	----
Power requirements: 3 Watt	----	----
Operating temperature: 0 °C to 50 °C	----	----
Weight: 15 grams	----	----
Communication cable: Basler cable USB3.0, Micro B sl/A, P, 3 m – data cable	----	----

The resolution is capable enough to provide an object resolution of 10.8 pixels per mm as per the chosen specifications. The object resolutions seem capable enough to detect minor cracks accurately. The camera is coupled with the 5 mm lens to achieve the desired field of view and working distance. Additional lights are used in the experimental setup to neglect the effect of ambient light conditions and maintain desirable uniform illumination over the object of interest. White led lights with 1500 lux are sufficient to meet the illumination requirements and even suppress the effect of shadow over the concrete cube surface.

Figure 12 shows the vision hardware used for image acquisition. Figure 12 (a) shows the industrial camera, Figure 12 (b) shows the USB3.0 communication cable, and Figure 12 (c) shows the lens and lights used for the development of the experimental setup shown in Figure 12 (d). The industrial camera comes with a board level wherein the CMOS sensor mounted on a PCB is not enclosed in a metal enclosure. A low-cost aluminium enclosure is fabricated to fit at the board level with a lens holder and opening for a communication cable.

Detection of cracks occurring on the concrete cube surface alone will not be sufficient to analyze the behaviour of the cube under compressive loading. The applied load should be acquired along with the images of the monitored surface during the testing. The compression testing machine had a feature of load data acquisition. However, proprietary software was required for the data acquisition and the sampling rate was much faster than the image acquisition rate. Hence, a separate 200 Ton compression load cell of pancake type is used. A signal conditioning unit, analog-to-digital converter and data communication unit are performed in the load cell controller. In order to acquire the load data from the controller via industrial communication, a serial converter is used. The load cell, controller and signal convertor also allowed to maintain synchronization in the data acquisition process. Both load data and image data are acquired in a common software platform and synchronization is sustained.



(a) (b)

(c) (d)

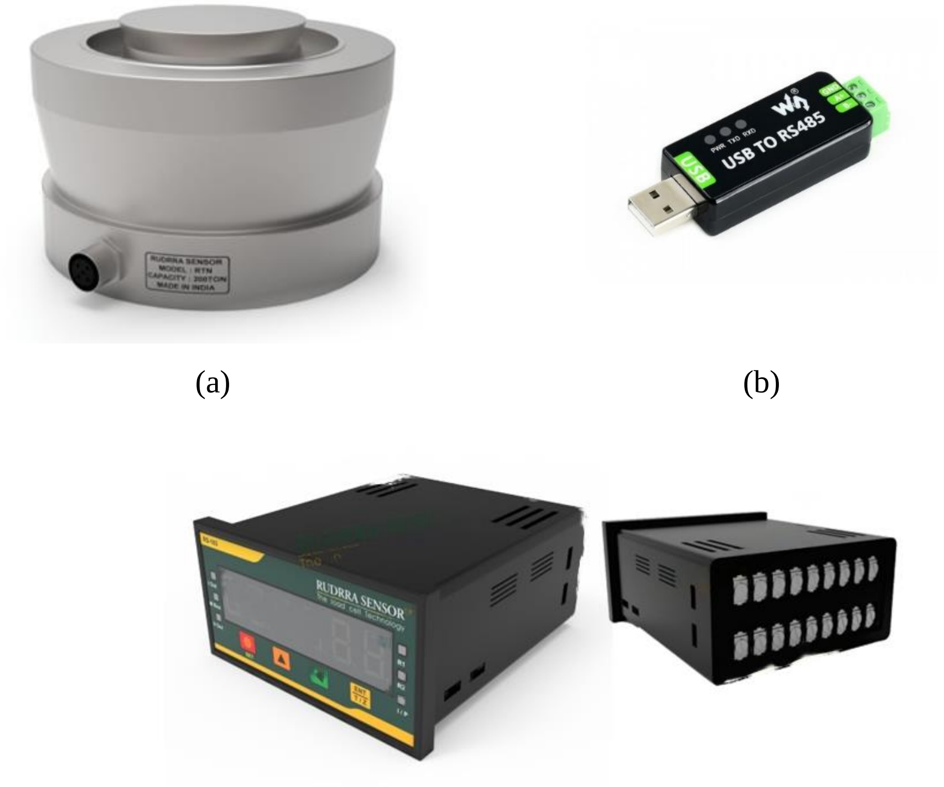
Figure 12. Vision hardware.

The load cell capacity of 200 Tons is chosen to fulfil the load measurement requirement of different strengths of concrete cubes. The equivalent load measurement capacity is 1992.80 kN. This capacity is sufficient enough to acquire the compression load of concrete cubes with high compressive strength. A load cell is a passive sensor which required external excitation and has very low output in terms of voltage. The specific requirement of a load cell signal conditioning, amplification, filter, analog to digital conversion, and data communication is provided by the load cell controller. The controller is a single device which can handle the conditioning, conversion and communication requirements of a load cell sensor. The controller has onboard digital indicators, re-calibration and parameter setting functions. The load cell controller can also communicate the raw data as well as calibrated data over Modbus Remote Terminal Unit (RTU) via serial RS485 communication. The data is sent to a computer system with the use of a USB to RS485 bidirectional converter.

Table 2. Instrumentation hardware specifications.

Load cell	**Load cell controller**	**USB to RS485 converter**
Model and Make: RTN C 200 T, Rudrra sensors, India	Model and Make: RS - 103 – 24, Rudrra sensors, India	Model and Make: Industrial USB to RS485 bidirectional converter, Waveshare electronics, China
Rate load: 200 Ton	Supply voltage: 220 V AC, 50 Hz	Host port: USB
MOC: Stainless steel	Panel mount	Device port: RS485
Rated output: 2.0 ± 0.1 % mV/V	ABS body	Baud rate: 300bps – 921600 bps
Combined error: 0.015 % of rated output	Dimension: 96 mm × 48 mm	Operating voltage: 5 V
Input balance: 775 + 5 Ω	Outputs: 16-bit Analog output, 2 relay outputs, Modbus RTU (RS485)	Host connector: USB A
Output resistance: 702 + 2 Ω	Display: 6-digit seven segment	Transmission distance: 5 meters
Insultation resistance: > 5000 Ω	Precision: 1/100000	Device connector: Screw terminal
Safe overload: 150 ± % of rated capacity	Speed display: up to 600 samples per second	Pins: A+, B-, GND
Ultimate overload: 300 ± % of rated capacity	ADC: 24-bit delta sigma	Led indicators: TXD, RXD, PWR
Recommended excitation: 5-12 V (DC or AC)	Filter: Digital filter	Temperature: -15 ° C to + 70 ° C
Protection class: IP67/IP68	----	Operating system: Mac, Linux, Android, WinCE, Windows 10 / 8.1 / 8 / 7 / XP

The specifications of the load cell, load cell controller and data communication module are mentioned in Table 2 (Low profile compression load cell (RTN) , 2022) (load indicator , 2022) (Industrial USB to RS485 Bidirectional Converter , 2022). Figure 13 shows the instrumentation used in the experimental setup. Figure 13 (a) shows the compression load cell, Figure 13 (b) shows the load cell controller and the USB to RS485 converter is shown in Figure 13 (c).



(a) (b)

(c)

Figure 13. Instrumentation hardware.

The image and load data should be acquired continuously during the compression testing. The testing takes around 1.5 minute to 3 minutes time depending on the strength of the concrete cube and the loading rate. In order to capture the data continuously, a stable and adjustable mounting of the camera and lights is necessary. Hence, an industrial aluminium profile section of 40 mm × 40 mm is used to fabricate an adjustable mounting stand to hold the camera and light. The mountings are designed in a way to support easy adjustments in horizontal, vertical and on-axis rotation. The assembly can be effortlessly kept for use in front of the compression testing machine during the testing. Once the testing is completed, the stand can be easily kept aside. The entire setup comprising vision hardware, instrumentation hardware and mounting assembly is covered in the system integration sub-section of the chapter.

A high-configuration computing system is required to meet the overall computing requirements of the experimental setup. The computation requirement comprises handling vision and instrumentation hardware, synchronous data acquisition, image processing, graphical visualization and data logging. A personal computer is used to meet the overall computing demands of the system. The computer has an intel core i7 – 9750H Central Processing Unit (CPU) with a clock frequency 2.60 Giga Herts (GHz), 6 cores, 12 logical processors, 16 Giga Byte (GB) physical Random Access Memory (RAM), 256 GB Solid State Drive (SSD), 1 Tera Byte (TB) Hard Disk Drive (HDD) and Windows 10 home version. An additional 4 GB NVIDIA Geforce GTX 1650 graphics card is available.

The regulated electric power supply is required for the operation of the electronic hardware used in the experimental setup. A single-phase AC supply is connected to lights and a load cell controller. The industrial camera receives power supply from the USB3.0 port of the computer system used in the setup. Similarly, the USB to RS485 converter received a power supply from the USB port of the computer system. The camera is used in the software trigger mode and hence input-output wiring is not required in the camera. The lights are kept on continuously during the experiment and hence strobe output for the light is not necessary. The load cell excitation and output signals are wired to the dedicated pins of the load cell controller. Modbus RTU signals of the load cell controller and connected with the signals of USB to RS485 converter pins A+, B- and ground.

3.3 Software selection

A software platform is always required to control and operate the hardware. The experimental setup comprises vision and instrumentation hardware which are controlled by software running on the computer system. The software should be selected and developed to perform the acquisition of images from the camera and load data from data acquisition in synchronization. In addition to that, the software should also have modules to allow image processing, data logging, and graphical visualization functionality. Training, validation and testing of AI and ML modules should also be available in the software.

Looking into the complete software requirements, LabVIEW software is chosen. It is a graphical programming environment by National Instruments, USA used by engineers, academicians, and industry professionals to develop automated research, validation, and production test systems (What Is LabVIEW? , 2022). In addition to the basic functionalities, LabVIEW encompasses various toolkits and modules like data communication, file IO, instrument IO, report generation, signal processing, vision acquisition, vision assistant and various others. The modules and toolkits are supported by addon modules which can be installed through the VI package manager. The availability of these functionalities makes LabVIEW a suitable choice for hardware integration. Above all, a graphical user interface is a part of the graphical programming in LabVIEW, which removes a separate need for the design and linking of the user interface.

AI and ML functionalities are sufficed by the use of Python and libraries like Keras, Tensorflow, OpenCV, Matplotlib, Scikit learn, Numpy, Scipy and others. These additional libraries are required for the AI, ML and other functionalities as python does not feature these modules. In addition to that, google Colab services are also used for model training. The experimental setup hardware is seamlessly integrated with LabVIEW and python to meet the defined objectives and scope of work.

3.4 System integration

Vision hardware, instrumentation, and software development using LabVIEW and Python should be integrated to perform crack detection and monitoring with respect to applied compressive load. The complete experimental setup comprises of compression testing machine of 3000 kN capacity, an industrial camera for capturing images during compression testing, a lens, lights, a load cell, a load cell controller, a USB to RS485 converter a computer for processing captured images and multiple samples of a standard concrete cube of size 150 mm × 150 mm × 150 mm used for testing. The camera resolution is 2592 pixels × 1944 pixels which are used to capture the surface image of a concrete cube of size 150 mm × 150 mm × 150 mm. A field of view of 180 mm width is considered to cover the entire cube in the image. A total of 1944 pixels per image are captured over a 180 mm field of view to achieve an object resolution of 10.8 pixels per mm.

The measurement accuracy of overall crack detection and crack measurements is affected not only by the camera specifications but also by the software algorithm and integration of hardware and software. The surface images of concrete cubes are continuously captured for each specimen along with the applied compressive load. For each compression test, the specimen is kept in the machine for testing and simultaneously the image and load acquisition hardware are adjusted and configured. Once the testing and acquisition setup is ready, the software is configured to carry out continuous and synchronous data acquisition, processing and storage. Post setup, the compression test is started. At the end of the compression test and failure of the specimen, the acquisition is stopped. Approximately 150-180 images are captured during the compression test of each specimen.

I

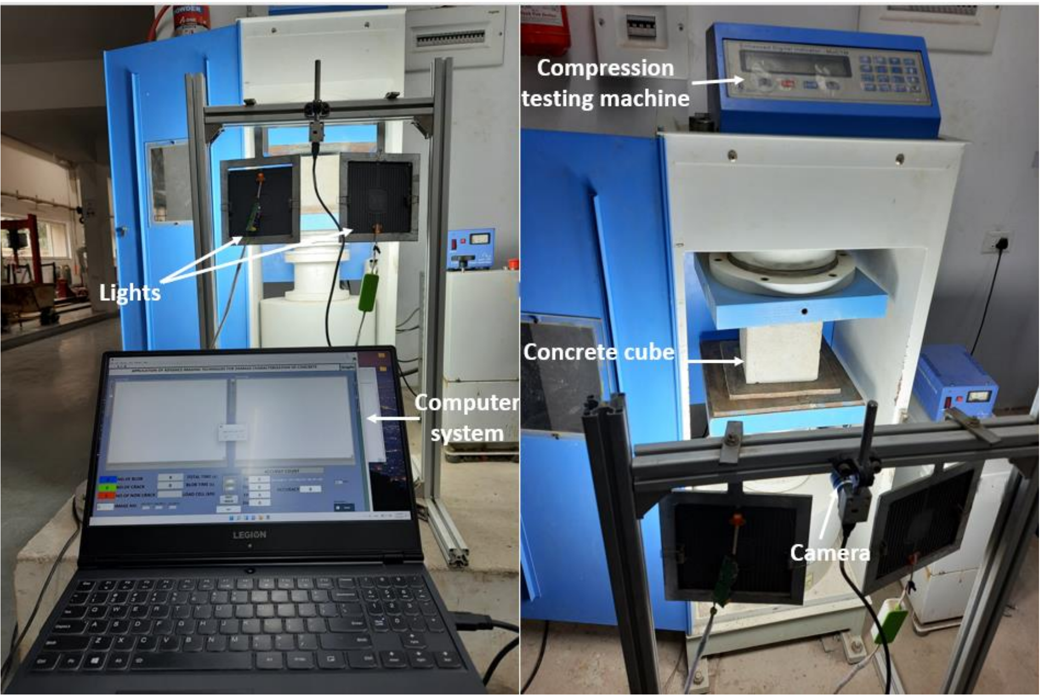


Figure 14. Experimental setup.

Figure 14 shows two views of the experimental setup developed to demonstrate real-time crack monitoring in synchronization of loading implemented during compression testing of concrete cubes. The camera, white lights, computer, compression testing machine, and concrete cube are shown with annotations and arrow marks. One of the prime requirements of image processing applications is a good resolution, uniformly illuminated and consistent images. Concrete cubes are greyish in nature and have a narrow difference in the intensities of cracks and random dents on the surface of concrete cubes. These dents may be detected as cracks by conventional image processing-based algorithms which consider greyscale intensity for segmentation and detection. The location and size of these dents, small cracks, and large cracks are random in nature. Hence, obtaining uniformly illuminated and consistent images is of prime importance looking into crack detection accuracy.

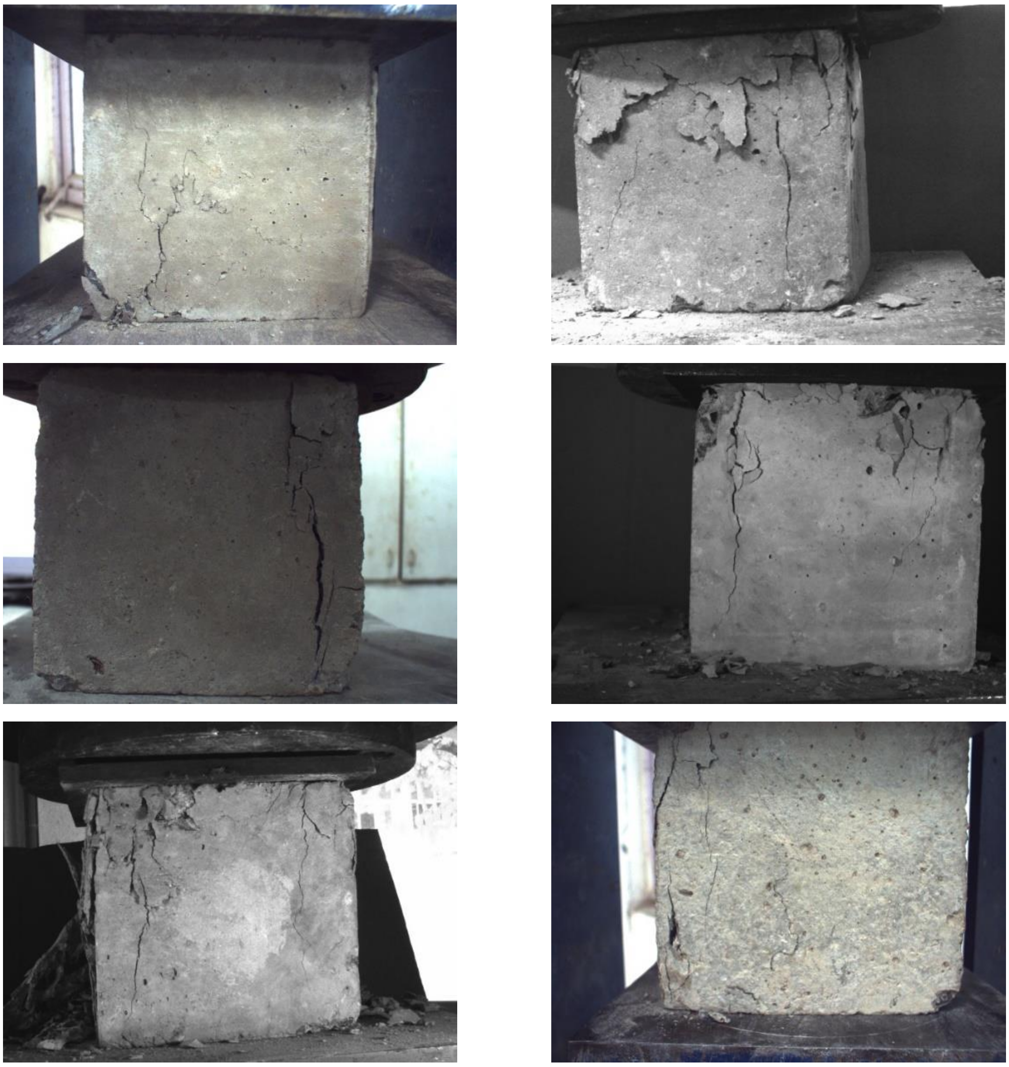


Figure 15. Concrete cube surface images without illumination and experimental setup.

Figure 15 shows the sample images acquired using the industrial camera with the lens but without illumination and experimental setup. These images lack consistency in terms of contrast, the improper orientation of the object, presence of shadow and background. All six images have a lot of background and the greyscale intensity is not consistent as well. In addition to that, the orientation of the concrete cube with respect to the edges is not aligned. The presence of shadow in the upper half of the cube is also evident in these images. These issues are visually evident and make the algorithm development extremely challenging and time-consuming. For an instance, threshold operation applied on these images with a common threshold value will result in variable segmentation. Hence, it is very difficult to accurately segment the crack and non-crack or background with conventional image processing algorithms.

Figure 16 shows the images captured with the experimental setup consisting of the illumination. The flexibilities of the acquisition setup leverage the user to acquire the image of the concrete cube with limited background, improved contrast and maintain the overall histogram of the image, well oriented. Further, an important advantage of using an industrial camera over a normal webcam or any other camera is that the camera acquisition parameters can be tuned to capture good images. Moreover, industrial cameras are capable to transfer images to the computing device on the fly and at a good frame rate. For an instance, an industrial camera can tune parameters like exposure time, white balance, gamma, saturation and various others. With webcams, these parameters are in auto mode which is unsuitable for such kinds of applications.

On visual comparison of acquired images shown in Figure 15 and Figure 16, it is evident that the images captured with the acquisition setup are consistent and well illumination as compared to the ones captured without setup. Orientation, illumination, background and greyscale intensity look quite consistent in all the images. No signs of shadow are visible in these images. These consistent images vastly assist in the dataset generation for CNN model training and improve the overall performance of the CNN model for accurate crack detection.

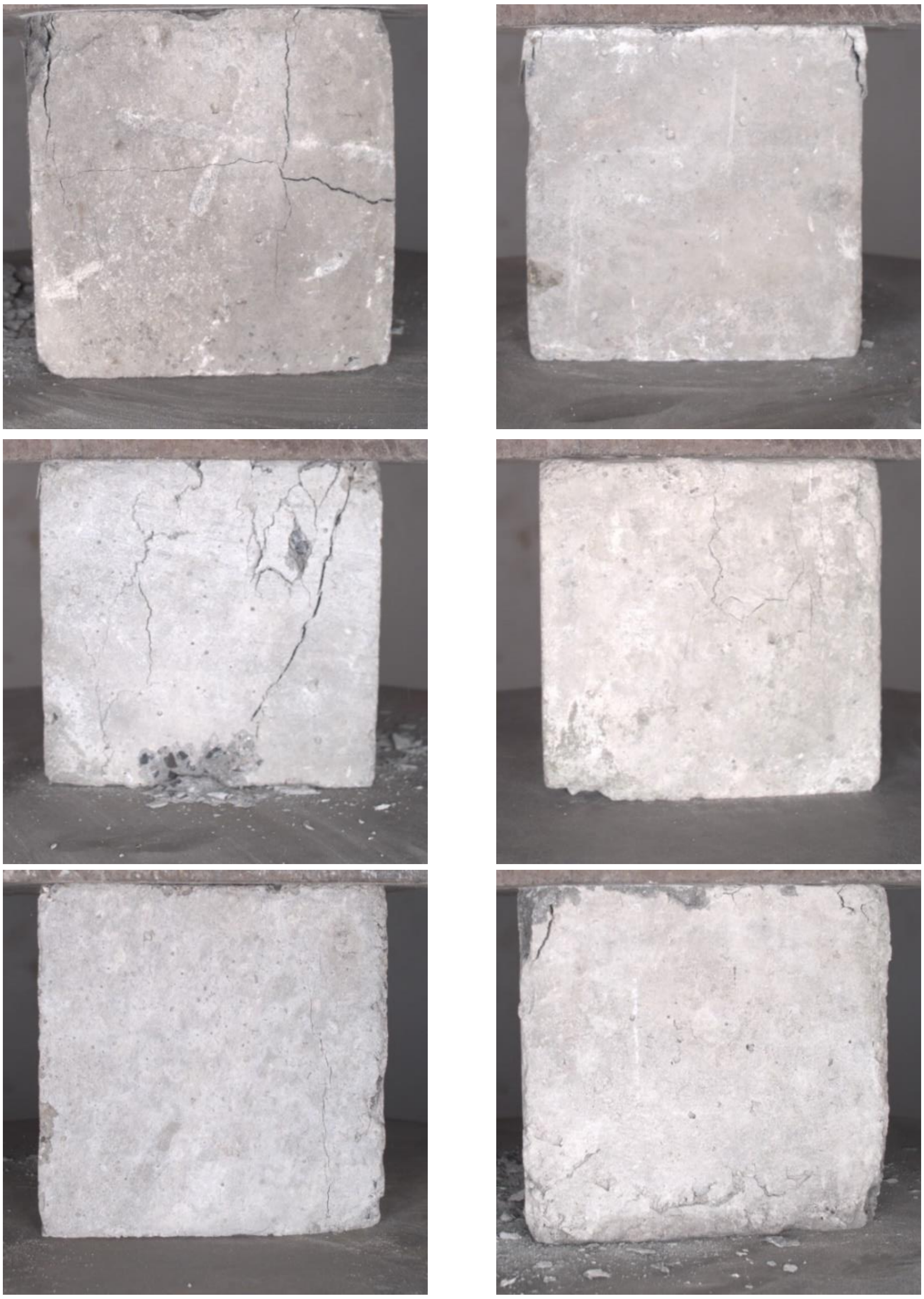


Figure 16. Concrete cube surface images with illumination and experimental setup.

A conventional image processing technique for crack detection would start to consist of functions like pre-processing, segmentation and classification. With ready-to-use tools and frameworks like TensorFlow, Keras, Caffe, Microsoft Cognitive Toolkit CNTK, PyTorch, MXNet, and Chainer building, the deployment of CNN has become fairly simple, fast, computationally affordable and efficient. The crack detection methodology implemented in the present work is discussed in chapter 4. A few sample images of multiple concrete cubes and load data are shown in the next sub-section of the chapter.

3.5 Concrete cube images and load data

One of the primary and important requirements of an image processing application is a good set of images. The images should cover all the possible variations in the object under inspection and other conditions. With the development of an experimental setup and selected vision hardware, the images are acquired. This would not only assist the crack detection methodology but also be useful for further analysis of concrete cube behaviour under compressing testing. The efficacy of the experimental setup is evident from the images shown in Figure 16. Various concrete cubes are tested with the use of a compression testing machine and the developed experimental setup.

The compression testing machine is initially set up conventionally by placing the concrete cube in between the plates for testing. Along with the cube, a load cell is placed for load measurement and the experimental setup consisting of vision hardware and instrumentation is set up appropriately. In addition to that, the software is set up and kept ready for data acquisition as soon as the compression testing is started. The data acquisition is carried out synchronously in order to acquire the surface image and applied load data at the same moment. For a concrete cube, if 185 images are captured and stored, equal numbers of load data are also acquired and stored in the memory. Keeping the same loading rate as described in the IS 516 standards, multiple concrete cubes of different compressive strengths are tested for compression strength. During the testing of a concrete cube, approximately 120 to 270 images are captured for each cube. The number of images may vary due to the difference in their compressive strength.

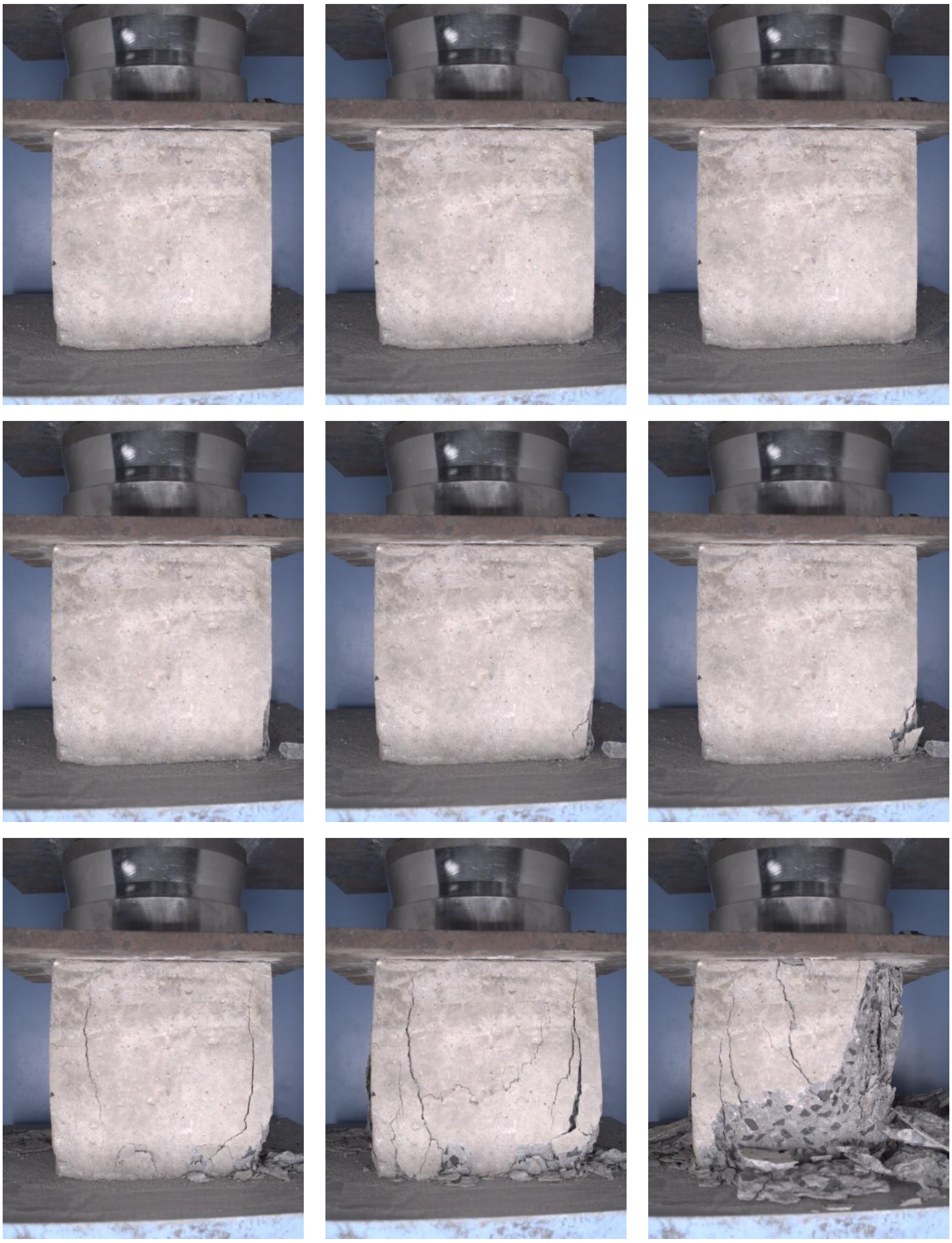


Figure 17. Sample images of a concrete cube captured during a compression test.

A cube is tested for compressive strength and 167 numbers of images and load data are acquired. In order to show the placement of the cube along with the place and load cell, the images are captured with a wider field of view. Figure 17 shows 9 images out of the total 167 images captured during the compression test cycle. Image numbers 1, 21, 41, 61, 81, 101, 121, 141, and 161 are shown in Figure 17. Cracks can be seen from image number 101 onwards and the right half of the concrete cube completely ruptures and spalling of the concrete surface is observed in image number 161. The load data collected for the same concrete cube with 167 load data points are plotted against time in seconds as shown in Figure 18. The peak load of the concrete cube is 797 kN.

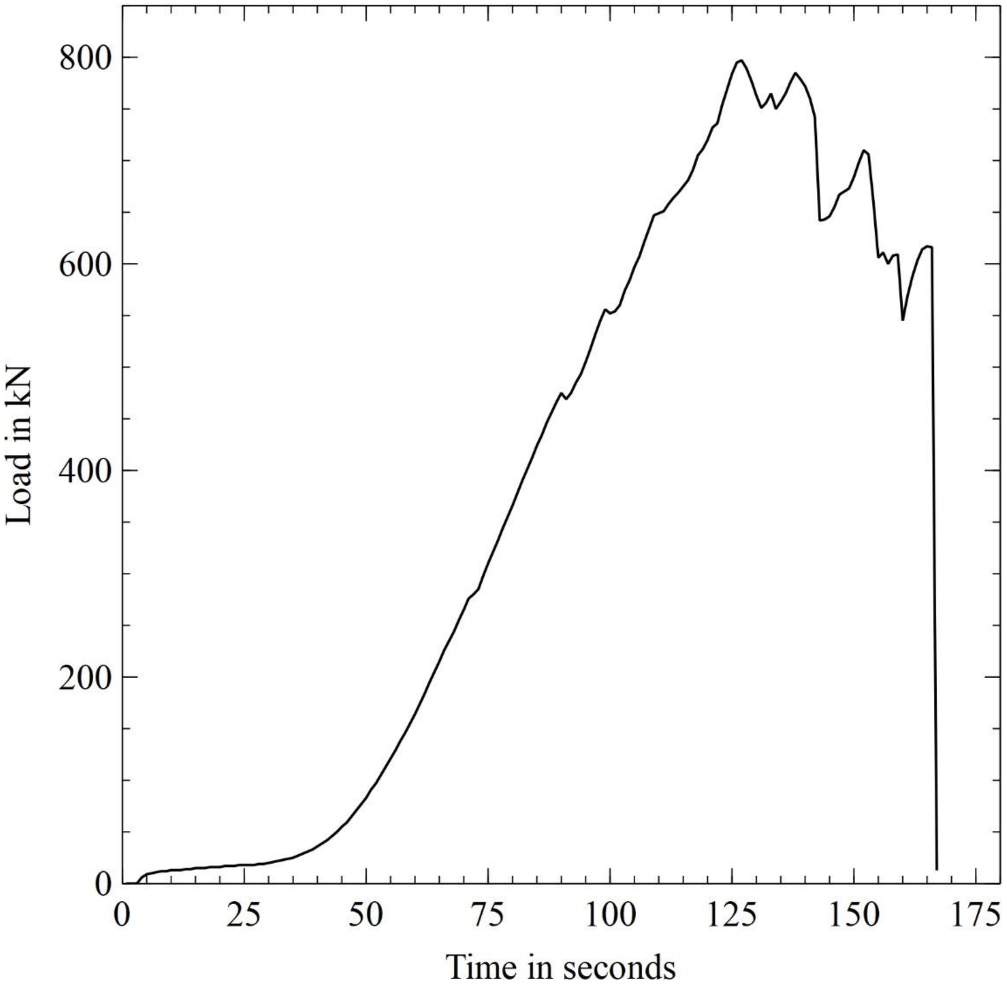
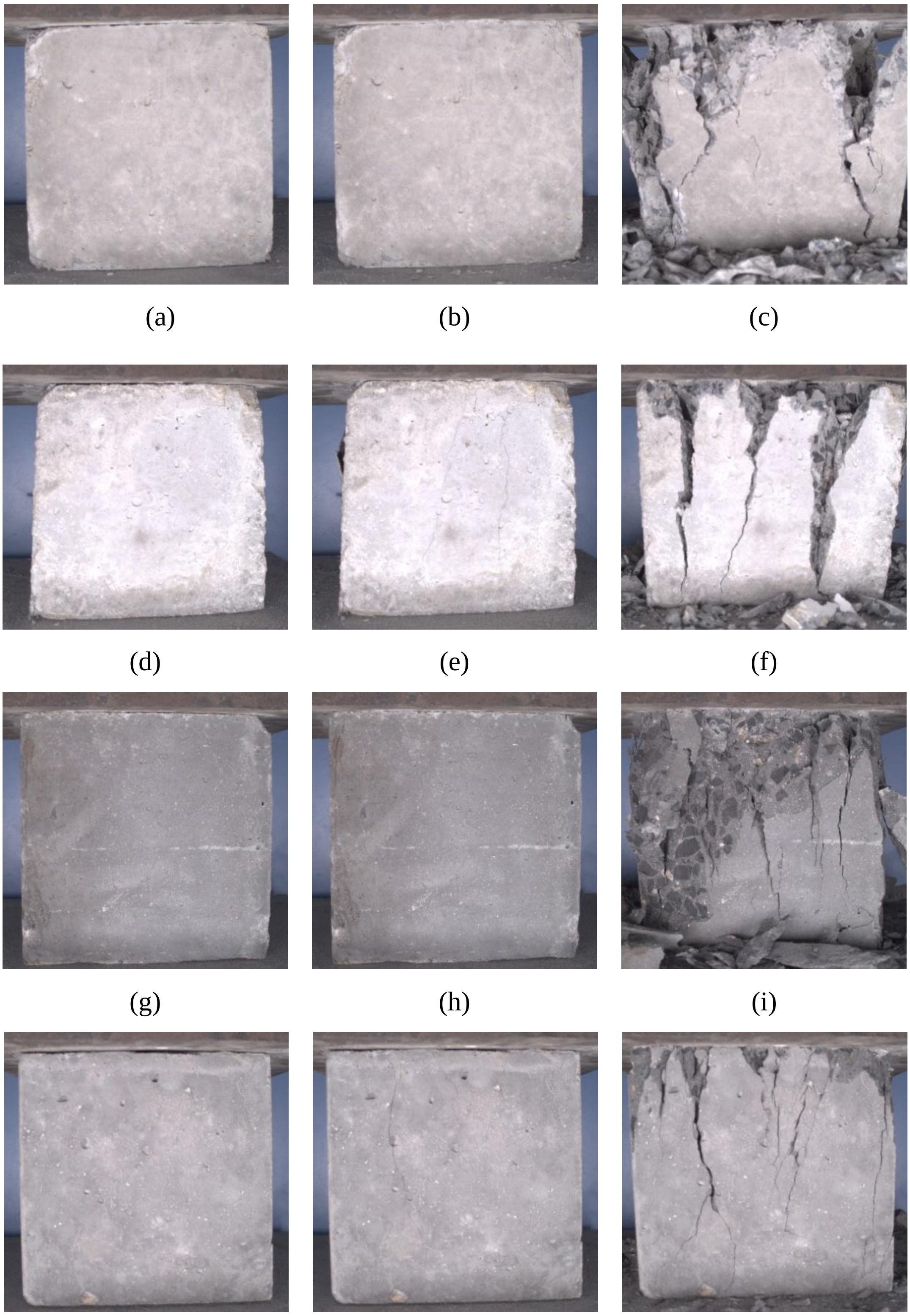


Figure 18. Load versus time graph of a concrete cube.

A few more samples of images are presented in Figure 19. Three images of each cube are shown in one row. The three images consist of the first image at minimum load conditions with zero cracks, the second image showing the concrete cube with the initiation of crack and the third image shows in the last image after the test. A total of 5 concrete cubes of data of images and load are presented in order to visualize the process of data acquisition and the advantage of developing an experimental setup.

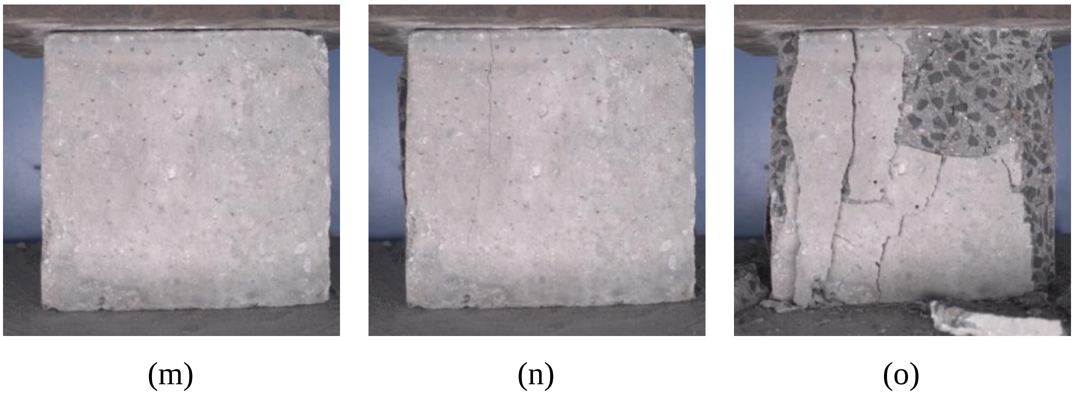


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)



(m) (n) (o)

Figure 19. Sample images of multiple concrete cubes captured during the compression test.

In few of the crack initiation images which are shown in the Figure 19 sub figure (b), (e), (h), (k), and (n) show the cracks are very minor and thin. These cracks are sometimes difficult to notice with the naked eye. In a few cases, these cracks occur at a corner and progress in a diagonal direction. In some cases, these cracks occur in the centre of the cube, start from the top edge of the cube and progress downwards. The third column shows the final state of the concrete cubes after the compression test cycle. The concrete cubes have completely deteriorated and spalling of a major portion of the concrete cube can be seen in some cases.

Figure 20 shows the load versus time graphs of these cubes. The load versus time plot shown in the figure belongs to cube number 1 whose images are shown in the sub figures (a), (b), and (c) of Figure 19. Similarly, load data of cube numbers 2, 3, 4, and 5 along with images are shown in Figure 20 and Figure 19 respectively. The total number of sample captures varied from 141 samples for cube number 1 to 194 samples for cube number 4. The minimum load is 0 kN whereas the maximum load is 1106 kN measured for cube number 5. The peak load for cube numbers 1, 2, 3, 4 and 5 are 338 kN, 1046 kN, 737 kN, 782 kN, and 1106 kN respectively. The total number of samples acquired for cube numbers 1, 2, 3, 4 and 5 are 141, 162, 152, 194, and 191 kN respectively.

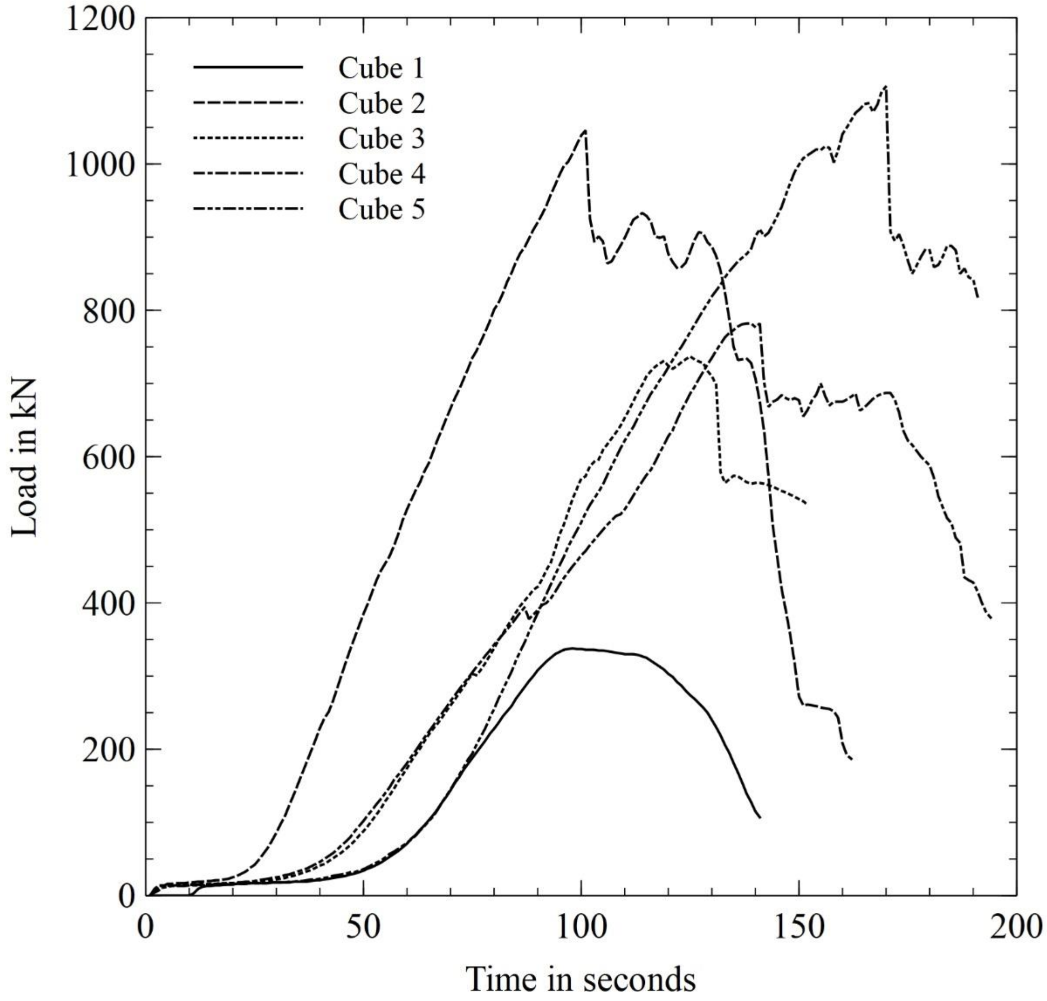


Figure 20. Load versus time graph of multiple concrete cubes.

The image and load data presented for multiple concrete cubes show the effectiveness of the integrated system and hardware, and software selected in order to achieve the specified objectives. However, data collection is the first primary step essential for crack detection. Looking at the images shown above, it is evident that the concrete cube surfaces are quite challenging to perform accurate crack detection. A novel crack detection methodology is developed in order to address the problem and the same is discussed in chapter 4.

Chapter 4 Crack detection methodology

Chapter 4 of the thesis covers the entire crack detection methodology which is developed using a state-of-the-art CNN based approach. A brief background of CNN and typical architecture is discussed in detail. A flowchart of dataset generation, data augmentation, and labeling is presented in the chapter. An in-depth discussion of the Inception v3 CNN model which was chosen for the classification of non-crack and crack contours is covered. Software platforms, libraries and the overall implementation approach were covered in the next part of the chapter. The chapter ends with the testing, result images and discussions.

4.1 Research methodology

The monitoring of cracks in a structure is carried out conventionally with the help of experts who manually monitor the cracks and extract or estimate their parameters like crack length, crack area, crack width and crack depth (Yao , 2014). Crack evaluation should be carried out based on these crack properties both in real-life structures as well as in laboratory environments. The study of crack properties in real-life structures can enable structural engineers to take appropriate action based on the extent of the damage. In the case of a laboratory environment, the properties can provide detailed and progressive information on concrete performance in different concrete testing with respect to applied loading.

Looking into the disadvantages of manual inspection methods, an automatic, continuous and accurate approach is explored in recent years by researchers. The disadvantages of manual inspection methods can be addressed by using the vision system and intelligent algorithms to address the challenges of crack detection and properties measurement (Yung-An Hsieh , 2020). In recent years, image-based crack detection has received a lot of attention. As an outcome, distinguished contributions have been observed in the domain (Dihao Ai , 2023). Majority of the applications addressed in past have focused on monitoring crack development in concrete structures without load monitoring. While the present research work addresses the real-time monitoring of crack development in synchronization with load acting on concrete structures. Assessment of load acting on structure along with monitoring of cracks helps in identifying the reason for cracking.

An experimental setup is developed in the present work to demonstrate the monitoring of crack developments with the measurement of loading in real-time. The need of automated crack detection and monitoring is addressed with the setup. In addition to that, robust crack detection methodology is research and developed in the work. Crack detection in concrete cube images captured during compression testing is carried out using a CNN and for load measurement load cell along with a data acquisition system is used. The setup is developed such that synchronization is maintained between the image and load data acquisition. The system is developed to carry out crack detection on surface images of standard concrete cubes during compression testing.

The testing work is carried out on concrete cube of size 150 mm x 150 mm x 150 mm. These specimens are tested by a compression testing machine after 28 days of curing. These concrete cubes are tested for compression in a compression testing setup. Throughout a complete compression test cycle, surface cracks in the concrete cube can occur at any time. A crack may start at a compressive load and can expand and propagate randomly. Multiple other cracks can initiate at a compressive load and propagate randomly throughout the compression test cycle. Cracks are generally darker as compared to their background and have lower pixel values. They may be discrete or continuous in nature. Some cracks may initiate discretely and get joined as the load progress or vice versa. For an image of a concrete cube surface, the area of the crack will be larger as compared to the area of noises, and dents. interference objects and much smaller as compared to the background area. Accurate inspection and measurement of the crack properties would produce reliable results and would assist in much better research and development of concrete technology and testing.

Figure 21 shows the crack detection methodology developed in the work. As seen in the Figure, the input to the crack detection methodology is the surface image of a concrete cube under compression test while the expected output is shown as the detection of crack and non-crack parts in the image. The cracks are shown with green coloured bounding box whereas the non-cracks are shown with blue coloured bounding box. Such a detection and classification are the expected outcome of the methodology.

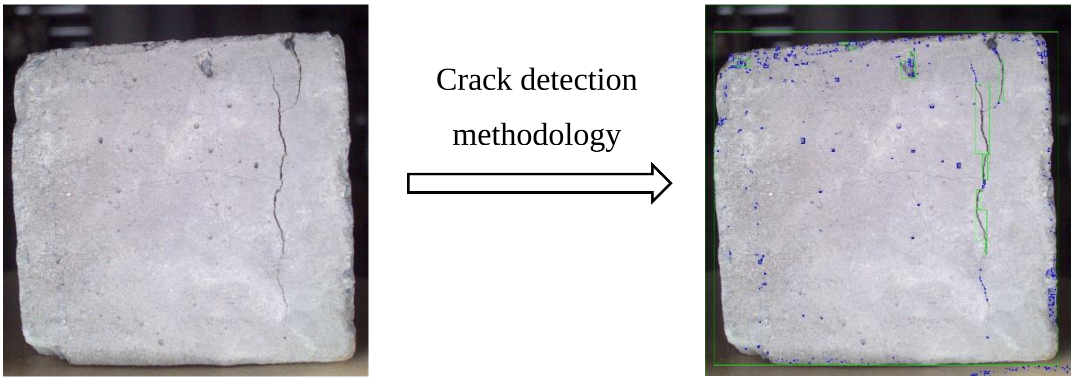


Figure 21. Crack detection methodology.

In addition to crack detection, the system is developed to perform monitoring and analysis of cracks with respect to applied compressive load. In order to develop real-time crack detection, an experimental setup comprising an industrial camera, load cell and other required hardware are developed. An integrated computing framework is developed using the NI LabVIEW based developed software along with a CNN model trained and developed using Python and additional libraries. The framework developed in the software performs image and load cell data acquisition, crack detection using CNN, crack and load monitoring, data logging and result display continuously.

In the case of a concrete cube surface image, the background colour is greyish in nature, while the crack is dark. The pixel intensities of cracks are much different from the pixel intensities of the background area but it matches the pixel intensities of noises, dents, blemishes, and other interfering objects present on the concrete cube surface. A critical challenge is to differentiate between an actual crack and noises, and dents. A conventional segmentation method like global thresholding or even adaptive thresholding does not provide consistent and accurate crack detection. Such methods rely on the difference between pixel intensity values of background and foreground or object of interest. These methods can easily segment background and crack, noises, and dents as foreground.

Online image data captured by an industrial camera will be transferred to LabVIEW along with real-time load data from a Data Acquisition System (DAQ). DL-based software will be developed and integrated with hardware which would enable the users to determine crack initiation, detect cracks, and measure their length and width. A graphical user interface will be designed to enable the user to visualize online images, crack detection, crack length, width and load. The outcome of the work will help in better analysis of concrete elements and would enable to gather of more accurate information about its behaviour in different conditions of loading.

In order to establish a justification and need for CNN-based crack detection, several conventional image processing and ML-based methodologies have been tried and tested. These approaches comprise pre-processing, image subtraction-based approach, percolation-based crack detection, and other additional methods like edge detection, gradient-based detection, using line filter and hessian matrix and others (Ahmed Mahgoub Talab , 2016) (Huang , 2010). Atsushi Ito et.al. proposed a crack analysis system for concrete block images. A high-resolution CCD camera was used to acquire crack images (Atsushi Ito , 2002). Shading correction, thresholding methods, crack tracking and labelling, feature extraction and analysis were applied to the acquired images and integrated into a tool. The authors used the sub-pixel interpolation method and a standard crack scale in order to measure the physical crack area. Two calibration methods were discussed to investigate the efficiency of the proposed method. Further, improvements in the crack area measurement were suggested.

Various approaches have been proposed in literature following which crack detection using such methods is used in the present work. Each of these methods is discussed in the coming sub-sections of the chapter while the testing results are discussed in the results and discussion section.

4.2 Image pre-processing

The primary objective of the pre-processing methods is to limit the amount of data for computation, improve the images to assist crack detection methods and remove noises and dents interfering with the cracks. In order to remove interference objects like noises/dents present on the surface under observation, filter pre-processing was used. Gaussian, averaging, median, and min-max filters are widely popular. Bilateral filters can preserve the strong edges while removing the noises. The intensity difference function in the bilateral filter only considers pixels with values equivalent to the central pixel value for blurring (Carlo Tomasi , 1998). Edges remain sharp due to large intensity variation but computation time is slightly higher as compared to other filters like the gaussian filter, median filter and others.

Image subtraction is used as a pre-processing method to remove noises and further enable a better differentiation between crack and non-crack objects. A synthetic white image is subtracted from the filtered image to obtain a subtracted image. The subtracted image removes the shadow present on the surface and provides a uniform illuminated image. Figure 22 shows the flow chart of image pre-processing operations applied along with the threshold-based image segmentation for crack detection. Images with respect to each of the pre-processing method are shown in Figures kept in the sub-topic 4.6 results and discussion.

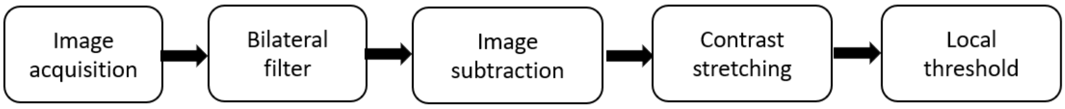


Figure 22. Flow chart of image pre-processing and crack detection.

Contrast stretching is used to increase the difference between the intensity values of cracks and dents. It is evident from the sample images seen in Figure 16 that the intensity values of the crack fall near to intensity values of the background, dents and noises due to the greyish surface of the concrete cube. The method proves useful as it enhances the contrast of crack pixels and suppresses the intensity of non-crack pixels in the images. In order to prove the use of pre-processing methods discussed above, it is necessary to perform crack detection.

Evaluation of the crack detection results with and without pre-processing should be carried out to justify the use of pre-processing operations. Hence, to carry out crack detection, a threshold operation is used. The operation will separate the crack and background portion of the image which can be further utilized to perform crack analysis and carry out measurement of crack properties. Traditionally, global thresholding like Otsu’s thresholding is deployed in cases of an image showing a bimodal histogram (Nobuyuki Otsu , 1979). However, as the concrete cube surfaces have random noises and dents which hinder accurate crack detection using global thresholding and due to the presence of shadows, adaptive thresholding is used. Niblack adaptive thresholding suits such a case and is hence used for accurate crack segmentation (Lalit Prakash Saxena , 2019). The results images of each operation used in the pre-processing of concrete cube images are discussed in the results and analysis section along with receiver operating characteristic analysis performed to justify the use of each pre-processing operation. After pre-processing the images are fed to the other crack detection methods which are discussed in the next section.

4.3 Conventional image processing

Image processing involves the processing of digital image data to extract important information and/or features. The extracted information can either be used for autonomous machine perception or human perception or storage or transmission. There are numerous operations, algorithms and methods available to extract the information from images. Similar methods have been presented in the literature work for the crack detection application (Pingrang Wang , 2010). Conventionally, crack detection is a problem of image segmentation where the primary task of the algorithm is to segment the crack accurately in any background conditions. Also, the algorithm should be robust enough to detect minor and micro-cracks and should not detect dents, noise and other unwanted parts present on the inspected surface.

Looking into the requirements of the crack detection algorithm, segmentation methods like edge detection, global and local auto thresholding, region growing, and other methods like image subtraction, structural similarity index, sobel gradient, and various others have been tried and tested. In addition to the pre-processing methods, segmentation methods are implemented to perform crack detection on the surface images of concrete cubes acquired with the use of the developed experimental setup. Each of the methods is discussed here in order to understand the challenges faced by conventional crack detection methods and justify the use of CNN.

4.3.1 Method 1 - Subtraction, line emphasis and iterative threshold

Image subtraction is an arithmetic operation applied on two images or between an image and a scalar value to obtain the difference image. The operation performs element-wise subtraction and generates an output difference image which may highlight the dissimilarities. Image subtraction is usually performed between two images that have substantial resemblances between them (Paranjape , 2010). In order to minimize the effect of non-uniform illumination, a background image is estimated from the input image.

In the present work, the subtraction-based operation can be very useful as the cracks appear on the surface of the concrete cube gradually during the compression test. Initially, at the start of the compression test, no cracks are present on the surface and gradually as the test progresses cracks start to appear and expand with respect to an increase in applied load. Hence, subtraction-based crack detection can be a useful methodology in such a scenario. Looking into the applications and importance of subtraction processing, Yusuke Fujita et. al. proposed work on crack detection (Yusuke Fujita , 2006). The authors utilized subtraction pre-processing, line emphasis pre-processing and iterative threshold. The results were evaluated using ROC analysis. Referring to this work, image subtraction and line emphasis filters along with iterative threshold are used for crack detection in concrete cube images. Figure 23 shows the flowchart of the method used for crack detection in surface images of concrete cubes.

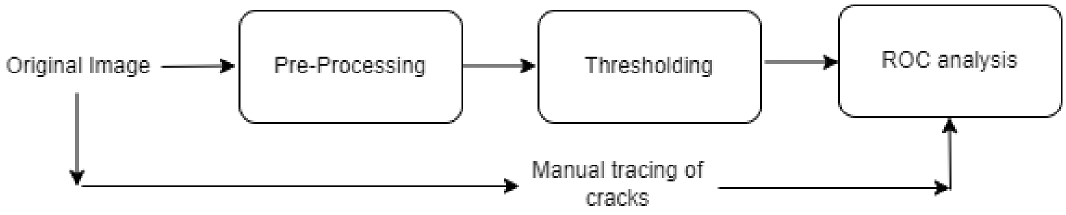


Figure 23 Flowchart of the crack detection method 1.

The method comprises of the masking operation, computation of the Hessian matrix, derivation of the eigenvalues and measurement of line symmetry. The masked image is obtained by subtracting the original image from the median filtered image. In order to compute the hessian matrix, second-order differentiation is calculated in the X and Y directions as shown in Equation (1). For each pixel in the image, eigenvalues are calculated using a 2 × 2 hessian matrix.

For an image I(x), the hessian matrix H is as follows,

$$H=\begin{bmatrix} I_{xx}(x) & I_{xy}(x) \\ I_{yy}(x) & I_{yx}(x) \end{bmatrix} \quad (1)$$

where partial derivative of the image I(x) are expressed as $I_{xx}(x) = \frac{d^2}{dx^2}$, $I_{xy}(x) = \frac{d^2}{dxdy}$, $I_{yy}(x) = \frac{d^2}{dy^2}$, $I_{yx}(x) = \frac{d^2}{dydx}$. Let the eigenvalues of H be $\lambda_1(x)$ and $\lambda_2(x)$ which represents the sharpness of the image. In order to distinguish the structures like blobs, and dents from the line, eigenvalues conditions should be satisfied. If | λ_2| >> | λ_1| ≈ 0, then the structure is a line whereas if λ_1 is negative and | λ_1| ≈ | λ_2| >> 0, the structure is considered to have properties similar to a circular blob-like structure. As disturbances and dents have similar structural appearances they are considered as one cluster. This helps the user to categorize the line structure easily. For λ_1 to be positive, its valuation is in the line path. The line structure is simplified based on three conditions, having the value of α between 0 and 1 as shown in the Equation (2).

$$\lambda_{12}= \begin{cases} |\lambda_2|\left(1+\frac{\lambda_1}{|\lambda_{2|}}\right) = |\lambda_2| + \lambda_1 & \text{, if } \lambda_2 \leq \lambda_1 \leq 0 \\ |\lambda_{2|}\left(1-\alpha\,\frac{\lambda_1}{|\lambda_{2|}}\right) = |\lambda_2| - \alpha\lambda_1 & \text{, if } \lambda_2 < 0 < \lambda_1 < \frac{|\lambda_2|}{\alpha} \\ 0 & , otherwise \end{cases} \quad (2)$$

The eigenvalues are calculated for every pixel using the hessian matrix. From the two eigenvalues obtained and depending upon the condition satisfied, the classification decision is made. Figure 24 shows the algorithm of the method.

Algorithm – Crack detection method 1 subtraction, line emphasis and iterative threshold

Step 1: Subtract smoothed image (median filter) from original image

Step 2: Compute Hessian matrix and eigen values

Step 3: Iterative threshold

Step 4: Comparison with manually traced cracks

Figure 24. Algorithm of crack detection method 1.

The method does not give satisfactory results when the contrast between the background and foreground object is not varying distinctively. For an image having a greyish background and uneven lighting conditions, the eigenvalues obtained partially satisfy the eigenvalue conditions mentioned above. As line emphasis is a point processing method, it works accurately for images with a whiter background. The testing and result images are covered in the results and discussion sub-section of the chapter.

4.3.2 Method 2 - Percolation based crack detection

Concrete blobs, stains and shading in several sizes make it difficult to distinguish cracks from the rest of the image. The use of conventional crack detection methods, which do not utilize the characteristics of cracks, increases the difficulty of segregating them. The use of percolation for crack detection was proposed by Tomoyuki Yamaguchi et. al. in their work (Tomoyuki Yamaguchi , 2008). The percolation model is an adaptation of the natural phenomenon of the permeation of liquid. The process of percolation starts from the seed point, from which it spreads to the neighbouring pixels according to the probability '*p*', which is used as a measure for ease of percolation. The seed point is known as the initial point for the start of the percolation process. From the neighbourhood, the pixel which exhibits the maximum probability '*p*', is percolated. By continuous iteration of this cycle, the region is percolated and reaches the edge.

The proposed method is local image processing based on the relation with neighbouring pixels. It is expected that only the linked pixels that are terms of intensity are percolated and form a cluster. If the seed point for the percolation process belongs to a connective region, the process takes place so linearly. Otherwise, if the seed point belongs to a background which does not have a linear feature, the process expands the percolated region two-dimensionally. Therefore, the characteristics of the seed point also define the shape of the percolated region in process of percolation. With the properties of cracks like circularity and length, the model becomes robust and highly accurate. The model of percolation is used to evaluate the focal pixel within the local region. For the given process, the input image with 256 greyscale levels is taken where the value of intensity is defined for 0 sets as black and 255 set as white. Figure 25 shows the flowchart of the percolation algorithm used for crack detection in surface images of concrete cubes.

In the proposed method, the model uses a 2-D square lattice. The basic model considers only four neighbourhoods; however, the improvised version considers all eight neighbourhoods. For the process, two different cases (Case I and Case II) are considered based on the subject the user focuses upon and the background. Case I considers the intensity of the focal pixel in the lattice is lower as compared to the background, the crack pixels are dark and others are bright. Whereas in Case II, the brightness of the focal pixel in the lattice is higher than the background, the crack pixel is bright and others are dark. Case I is considered in the present work as the intensity of crack pixels in concrete cubes is comparatively lower than the intensity of background surface.

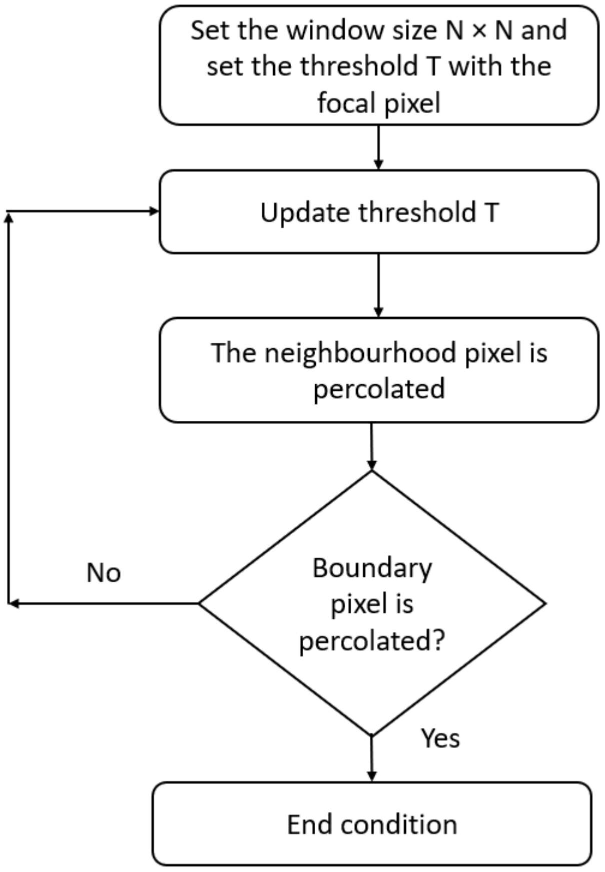

Set the window size N × N and set the threshold T with the focal pixel
Update threshold T
The neighbourhood pixel is percolated
Boundary pixel is percolated?
No
Yes
End condition

Figure 25. Flowchart of percolation process.

Figure 26 shows the algorithm of the percolation based crack detection method. For the model of percolation to run over an image, firstly a local N x N matrix is fixed and the maximum size of the window is set to M x M. The seed point is set as the focal pixel of the local matrix. The initial pixel is set as 'p_s' for the percolation process and the percolated region is classified as 'D_p'. The threshold 'T' is set as the initial pixel's intensity value. The Threshold 'T' is set to continuously update upon each iteration using the Equation (3).

$$T = \max(\max(I(p)), T) + \omega, \text{ where } p \in D_p \; (3)$$

Here ω is the accelerating parameter for the percolation model.

Algorithm – Percolation based crack detection

Step 1: Fix initial window of size N x N. The maximum window is defined as M x M. Assign the centre of the local window as the initial pixel p_s for percolation. D_p is percolated region and the percolation threshold T is set to the value of the initial pixel brightness I of p_s.

Step 2: Update the threshold T

Step 3: Dc is candidate region and comprises of the eight neighbouring regions of D_p. Check Dc and percolate the pixels whose brightness is lesser than the threshold T and included them D_p. If there are no such pixels, only the least bright pixel in Dc is included in D_p.

Step 4: If D_p reaches the boundary of the N x N window, the percolation operation proceeds to step 5, and N is incremented to N + 2 else the process goes back to step 2.

Step 5: Update the threshold T in a similar manner as step 2.

Step 6: In the neighbouring regions D_c of D_p, check the pixels whose brightness is less than T and included them in D_p. If there are no such pixels, the percolation process is terminated.

Step 7: If D_p reaches the boundary of the N x N window, N is incremented to N + 2. Otherwise, the process goes back to step 5.

Step 8: If N is larger than the maximum window size M, the process is terminated. Otherwise, the process goes back to step 5.

Figure 26 Algorithm of percolation based crack detection.

When the seed point is percolated, the eight neighbourhood pixels are taken into account and are listed as candidature region D_c. Each of the candidate region is checked and the pixels whose intensity value is lower than 'T' are percolated and included in D_p. When the percolation process reaches the boundary of the N x N lattice, the value of N is increased by 2, otherwise, the threshold modification step is repeated. Again, in the neighbouring region D_c of D_p, the pixels of intensity lower than the value of 'T' is included in D_p. Else, the value of N is increased by 2.

When the value of N increases than the maximum window size of M, the process is terminated. The properties of a crack are compared to check whether the percolated region is a crack or not. The circularity factor of the region D_p is checked to confirm that the percolated region is a crack. The value of circularity F_c is estimated as a characteristic of D_p and the formula is shown in Equation (4).

$$\mathrm{Fc} = \frac{4}{\pi}\frac{C_{count}}{C_{max}^2} \quad (4)$$

Here, C_{Count} is the number of percolated pixels in the region D_p. C_{max} is the maximum possible length of the region D_p. When the calculated F_c is close to 0, the shape of the percolated region D_p is linear and when the value is close to 1, the shape of the percolated region is circular. The updated accelerating parameter *ω'* for the model can be represented as a product of circularity F_c and accelerating parameter *ω*. The formula is shown in Equation (5). The tolerance is the value is the variance to the threshold taken as input.

$$\omega' = F_c.\ \omega \quad (5)$$

The threshold calculation is updated using the formula shown in Equation (6).

$$\mathrm{T} = \max(\max(\mathrm{I}(p)), \mathrm{T}) + \omega' \quad (6)$$

The stated percolation algorithm is used for accurate crack detection in the surface image of a concrete cube under a compression test. The result images and observations are discussed in the results sub-section of the chapter.

4.4 Machine learning

Strong contribution to the area of crack detection using ML, DL, ANN and fuzzy logic is discussed in the literature review section of the thesis. Drawing conclusions from the literature, it is observed that initially researchers have used conventional and derived image processing methodologies to perform crack detection till 2015-16. Post that, AI-based crack detection and damage assessment methods have become widely popular amongst the research community (Christian Koch , 2015). Detection of cracks found on roads, building walls, tunnels, heritage monuments, nuclear vessels, pavements, bridges and various other infrastructures have been addressed using AI-based methods.

The current problem of crack detection in concrete cube surface images captured during the compression test cycle also suffers from the stated limitations and challenges. The previous methods of line emphasis and percolation when applied to the concrete cube images, they suffer from the issue of similarity in the intensity level of dents, and noises with that of cracks. And hence segment the dents and noises present on the surface along with cracks and miss minor cracks in a few cases. Looking into the issues, the problem of crack detection is upgraded to the classification of cracks and non-crack parts post-segmentation. The non-crack parts are noises, dents, blemishes and other unwanted regions present on the cube surface apart from cracks. In order to mitigate all these issues associated with accurate crack detection in the concrete cube, an experimental setup is developed. The efficacy of the developed setup is discussed in chapter 3 of the thesis. The use of setup enables the acquisition of blur-free, consistent, uniformly illuminated and properly oriented images of concrete cubes under compression test. Hence, challenges associated with image acquisition are addressed with the help of experimental setup and vision hardware utilized in the setup.

Now the focus is shifted towards an accurate classification of crack and non-cracks parts based on the ML method. Since the intensity-based operations are not able to separate the cracks and non-cracks accurately, feature-based classification using a ML approach is undertaken. Using knowledge-based features of crack and non-crack parts, classification problems can be addressed with the use of a ML methodology.

As proposed by Suguru Yokoyama et. al., a ML -based approach can be found useful for crack detection (Suguru Yokoyama , 2017). ML methods are separated into supervised and unsupervised categories. Different methods like regression, classification, clustering, dimensionality reduction, and others are very useful and widely used in numerous industrial, commercial and research applications.

In order to address the present crack classification approach, K Nearest Neighbours (KNN) and Support Vector Machines (SVM) methods of supervised ML category are used. The model training requires feature data of both crack and non-crack parts. Pre-processed and binarized result images of multiple concrete cubes are used for feature extraction and data preparation. Crack and non-crack parts are viewed as regions in the image and the properties of each region are identified to collect the data for supervised training. The extracted features are bounding rectangle diagonal, perimeter, maximum ferret diameter, equivalent rectangle diagonal, area, elongation factor, compactness factor and Heywood circularity factor. The factors are chosen based on the features difference between cracks and non-cracks parts. Various other features of cracks and non-cracks were also tried using different images before selecting the mentioned features. Figure 27 shows the flowchart of the training KNN and SVM ML models.

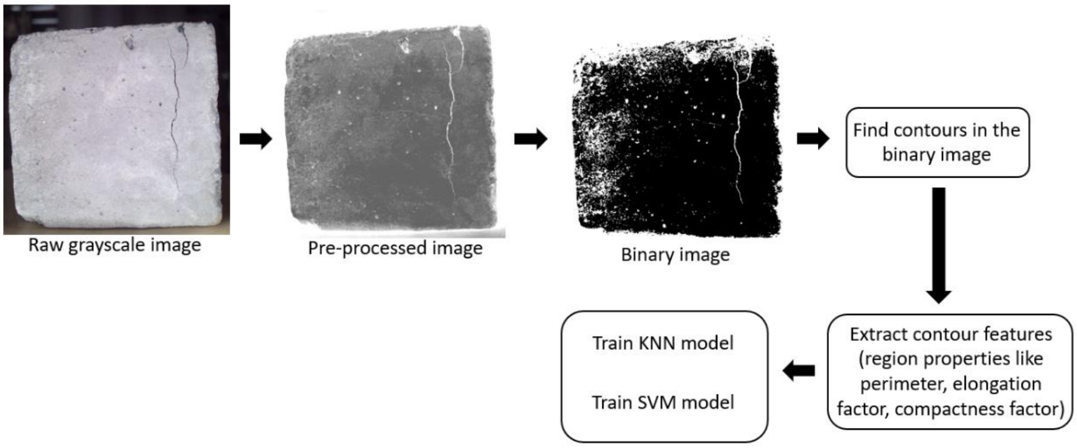

Raw grayscale image
Pre-processed image
Binary image
Find contours in the binary image
Extract contour features (region properties like perimeter, elongation factor, compactness factor)
Train KNN model
Train SVM model

Figure 27 Flowchart of training KNN and SVM models.

Cracks are looked at as line-like structures while the noise, dents and other non-crack parts are looked at as more circular and blob-like structures. And hence, the features are focused on shape-based features in order to distinguish between crack and non-crack. These features are extracted using the particle analysis tool of NI Vision Assistant. The particle analysis tool allows a user to select the properties of the region and view these properties for each region found in the binary image. These features can also be extracted in a Comma Separated Value (CSV) file. Post exporting the selected data in the CSV file, each feature row is manually marked as crack or non-crack by adding one more column. The features of cracks are marked as 1 whereas the features of non-cracks are marked as 0. A total number 485 number of data is collected using the particle analysis tool, out of which 255 numbers of data are belonging to the crack class and the remaining 230 numbers of data belong to the non-crack class.

The prepared dataset is used for training and testing the KNN and SVM algorithms utilized for crack detection. The problem under consideration is a binary classification and looking into the characteristics of the extracted features, KNN and SVM were chosen for implementation and testing. SVM is based on statistical approaches and tends to work best when the dataset is small in numbers and complex in nature. While KNN is a non-parametric approach and does not rely on the underlying data. KNN on the other hand identifies similarities between the new data and trained data in order to classify the new data. Hence, both KNN and SVM are justified for their use in the crack detection problem under consideration. The training and testing results of both methods along with performance analysis and crack-detected images are discussed in the results and discussion sub-section of the chapter.

4.5 Convolutional neural network

DL refers to the process by which computers can imitate human visual recognition behaviour. It uses different forms of neural networks, which are multi-layered networks inspired by the human brain's structure. One of the most popular image classification algorithms used in DL is the CNN. It uses artificial neurons, which are mathematical functions that measure the weighted number of multiple inputs and outputs an activation value. The conventional image processing method does not work well in the detection of cracks in concrete structures as the intensity range of cracks and non-cracks in the concrete block is almost similar.

The recent literature review presented by various researchers also claims that for the problem of crack detection, CNN-based segmentation performs better as compared to the conventional image processing-based methods (Jianghua Deng , 2022) (Narges Kheradmandi , 2022) (Younes Hamishebahar , 2022) (Dorafshan and Robert J. Thomas , 2018) (Md Zawad , 2021). Further, as the load increases on a concrete specimen, the area of the crack increases and it merges to become a longer crack so, area-based filtering of the non-crack block is inaccurate. CNN improves this task by combining the output of different trained convolutional networks in a fusion multilayer perceptron and the features present in the feature map extracted by the convolution layer are summarized by the pooling operation.

4.5.1 Overview

CNN, also known as ConvNet, is a DL algorithm, which mainly finds their applications in visual imaging tasks. It can take an image as input and can identify and/or classify various objects present in the images based on the features. A typical CNN consists of convolution layers, pooling or sub-sampling layers, activation functions and fully connected layers. A simple 4-5 layers networks to complex 40-50 layers networks are developed to solve difficult and visually challenging applications of the computer vision domain.

The overall architecture of the network is divided into feature extraction and classification part. The combination of multiple convolution layers, pooling layers and activation functions work for feature extraction whereas the fully connected layers along with the softmax activation function performs the classification task. A typical CNN architecture with respect to crack detection is shown in Figure 28.

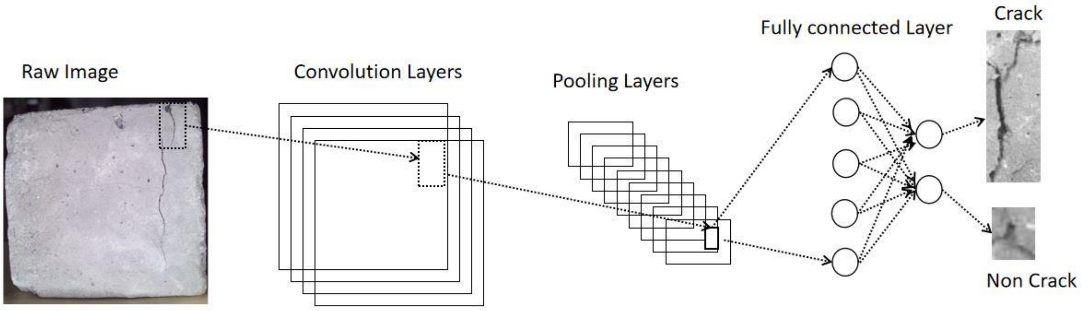


Figure 28. Architecture of convolutional neural network.

The CNNs are trained in a supervised learning mode where the labelled data is fed to the networks for training. During the training of the networks, each of the images is passed through a forward pass in order to calculate the loss and backpropagation to learn the optimized values of parameters by minimizing the loss. In the case of neural networks, the parameters which are optimized are weights and biases. Particularly for CNN, the values of a convolutional kernels are considered as the weights and biases.

These kernel and their optimized values are essentially responsible for automatic feature extraction. Backpropagation is an algorithm which is used in these networks to obtain the best-optimized parameters. These optimized parameters minimize the loss function. Gradient descent is the optimization algorithm which works to optimize the parameters to minimize the loss function.

For a CNN application, the typical solutional methodology includes sequential selection and execution of operations (Elgendy , 2020). The solutions start with defining the performance metrics and design or selections of baseline models to start with. Dataset collection and data preparation for training are the next important steps in the structuring of the application. Datasets are readily available for a lot of applications. If the dataset is not openly available, then the researchers have to put considerable effort and invest time to collect and prepare the dataset. Model training, evaluation and interpretation of the performance are the next steps. Based on this the model can be improved further by improving the hyperparameters associated with the network. Regarding the crack and non-crack classification applications considered in the current work, each of the sequential steps is discussed further.

4.5.2 Performance metrics

Accuracy, confusion matrix, precision, recall, and F score are commonly used performance measurement parameters with reference to classification applications. Defining the metrics for performance evaluation at the start is necessary to understand the goodness of the system. The problem under consideration is a binary classification problem addressed to predict a contour as crack or non-crack. Multiple contours are found in each image which is predicted as crack or non-crack contours. To examine the performance of the methodology, manual testing of crack detection is performed on the multiple images which are not utilized for the training, validation or testing of the network. The parameters used for the evaluation are a standard in CNN-based object detection and classification applications (Ali Raza Asif , 2020) (Chaitanya Nagpal , 2019) (Zewen Li , 2021). In the present study, crack is considered a positive class whereas non-crack is considered a negative class.

The number of crack contours found in the image is shown by parameter C whereas parameter NC indicates the number of non-crack contours. Both these numbers are manually calculated for each image and considered to be the ground truth or reference. The TP (True Positives) indicate the number of crack contours correctly predicted as cracks by the model. The TN (True Negatives) shows the number of non-crack contours predicted as non-crack by the model during the testing. The parameter FP (False Positives) show the number of non-crack contours incorrectly classified as crack by the model whereas the FN (False Negatives) indicates the number of crack contours incorrectly classified as non-crack by the model. The ideal values of FP and FN for an accurate CNN model should be zero. Accuracy is represented by A and it indicates the overall effectiveness of the CNN model. Accuracy can be calculated by taking the ratio of the total number of crack and non-crack contours classified correctly to the total number of contours shown in Equation (7) (Mahbub Hussain , 2018).

$$Accuracy\ (A)\ =\ \frac{TP + TN}{TP + TN + FP + FN}\ (7)$$

The parameter precision and recall represented by P and R respectively show the goodness of the network to classify the classes in terms of sensitivity and specificity. Precision refers to the ratio of the correctly classified number of crack contours to the total number of contours classified as cracks as shown in Equation (8) while recall refers to the ratio of the correctly classified number of crack contours to the total number of crack contours as shown in Equation (9). For an ideal network, the values of precision and recall should be 1 and the value of FP and FN should be 0. In order to summarise the classification performance of a model with a single metric which represents both precision and recall the F-score can be used. F-score is often called the harmonic mean of precision and recall and is shown in Equation (10).

$$Precision\ (P) = \frac{TP}{TP + FP}\ (8)$$

$$Recall\ (R) = \frac{TP}{TP + FN}\ (9)$$

$$F - score\ (F) = \frac{2{*}P{*}R}{P{+}R}\ (10)$$

Performance evaluation and interpretation observation are drawn using these parameters. Defining these parameters is essential as it will guide the approach towards model improvements.

4.5.3 Baseline model selection

A baseline model is a CNN model which can be used to start the training and further evaluation process. Initially, a simple or well-known CNN model architecture can be used to start. The architecture can be appropriately updated based on the observations and interpretations carried out from the performance evaluation. The number of layers, network depth, layer depth, pooling type, activation function, optimizer, and regularization are the important variables to be decided at the beginning while selecting the baseline model.

AlexNet, LeNet, GoogleNet, MobileNet, ZefNet, VGG16, ResNet50, MatConvNet, and DenseNet are some of the popular model architectures which are used widely for different vision-based applications (Alex Krizhevsky , 2012) (Jia Deng , 2009) (Andrew G. Howard , 2020) (V. V. Christian Szegedy , 2016) (Karen Simonyan , 2014) (Kaiming He , 2016) (A. Vedaldi , 2015). These popular CNN models, new customized network model architectures, novel methods, and integrated approaches are being developed for vision-based crack detection for different structures.

In order to select a particular baseline model out of the available options, the candidate model's performance on ImageNet Large Scale Visual Recognition Competition (ILSVRC) is considered (Laith Alzubaidi , 2021) (Dulari Bhatt , 2021). Ideally, the model showing the highest accuracy should be chosen. However, the model's capability to extract features and generalize on test data is also important. In addition to that, training resources available, time and memory limitations during inference are also considered in the selection process.

The Inception v3 network has 42 layers which include multiple convolution layers, pooling layers and fully connected layers. The network was the first runner-up in ILSVRC in 2015 (W. L. Christian Szegedy , 2015) (Suriani Mohd Sam , 2019) (Shane Barratt , 2018). The previous version of inception v1 also known as GoogleNet emerged as the winner of the ILSVRC competition in 2014. Achieving high-level accuracy with the decreased computational cost is the core aim of the GoogleNet architecture. Similarly, inception v3 also has a lower error rate, faster convergence, and reduced computation time. The model can classify an object from 1000 classes and was trained on 1000 images in each of the 1000 classes with a total of 1 million images for training, 50,000 images for validation and 1,00,000 images for testing purposes (Tsang , 2019). Inception v3 models were based on the concept of factorization wherein the convolution operations were factorized to reduce the number of connections/parameters without decreasing the model efficiency. It has been used in different image classification applications like flower classification (Xia , 2017). Considering these parameters and performance, inception v3 is selected as the baseline model to start the training of the dataset.

4.5.4 Dataset preparation

Dataset preparation is the next step in the sequence of a DL application. It involves image data acquisition, pre-processing the data and dataset splitting for training, validation, testing and evaluation. Figure 29 shows a generalised flowchart of crack and non-crack dataset generation. The detailed process of dataset generation is discussed further.

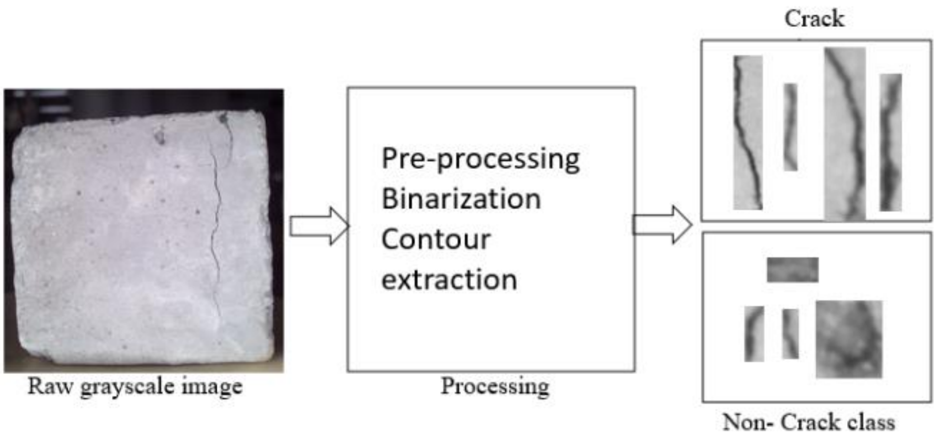


Figure 29. Flowchart of dataset generation.

The majority of the applications developed using CNNs have ground truth images or known labels or classes for supervised learning. These digits images are used for training, testing and validation of the CNNs. For concrete crack detection, ground truth images are not so commonly available. Also, the images for laboratory-scale concrete cubes are not available. Moreover, concrete cracks are of various sizes and shapes and have good intensity variations as well.

CrackNet is a CNN model applied for crack detection on road surface images (Allen

Zhang , 2018). Another CNN called CrackNet-V has been deployed for pixel-level crack detection on 3D asphalt pavement images (Yue Fei , 2019). To the best of our knowledge, CrackNet is not used for crack detection on a concrete surface. In addition to that, no particular CNN model has been widely researched.

In the present work, the development of the CNN application starts with data set preparation. With increasing applications of structural health monitoring in a variety of areas, CNN models like CrackNet will be developed solely for concrete crack detection. Even the dataset of concrete cube crack images is not openly available as of date. Structural Defects NETwork (SDNET) 2018 is available on Kaggle (Structural Defects Network (SDNET) 2018 , 2022) Sattar Doragshan et. al. presented this SDNET 2018 dataset in their work (Sattar Dorafshan , 2018).

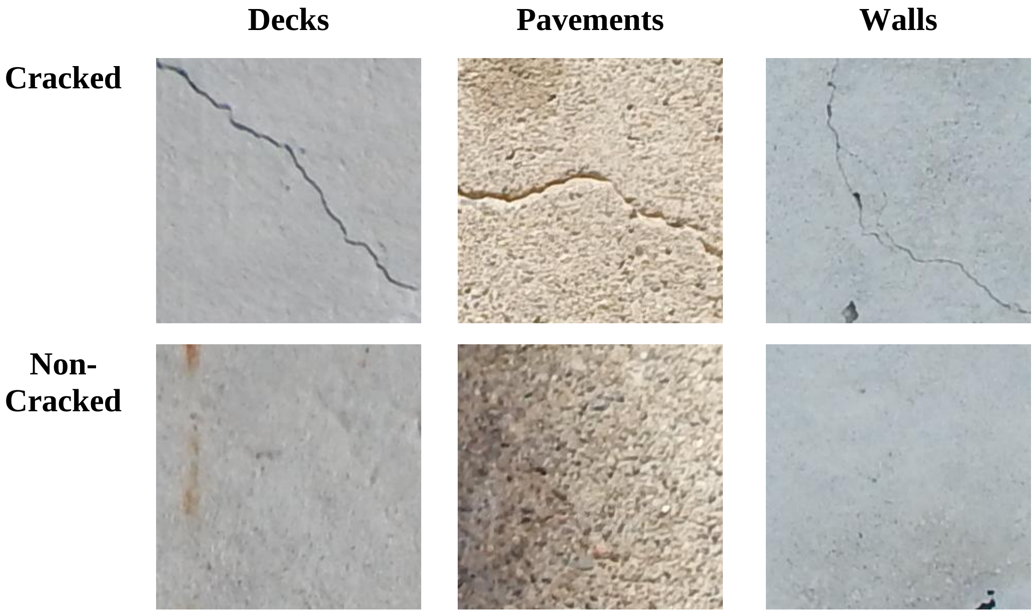


Figure 30 Sample images from SDNET dataset. (Structural Defects Network (SDNET) 2018)

The dataset was created for training, validation, and benchmarking of AI-based crack detection algorithms for concrete. It contains 56,000 images of cracked and non-cracked concrete bridge decks, walls, and pavements. A total of 230 images of cracked and non-cracked concrete surfaces are captured using a 16 MP Nikon digital camera. Out of the 230 images, 54 images from bridge decks, 72 images of walls, and 104 images of pavements were collected. Sample images of the dataset are shown in Figure 30.

The bridge decks were located at the Utah State University (USU) System, Material, And Structural Health (SMASH) laboratory. The inspected walls belong to Russell/Wanlass Performance Hall building located on the USU campus while the pavement images were acquired from the roads and sidewalks on the USU campus. Each image is segmented into 256 × 256 pixels sub-images of crack and non-crack parts. Each sub-image is manually labeled as 'Cracked' and 'Non-cracked' depending on the presence or absence of a crack in the image. The dataset includes cracks as narrow as 0.06 mm and as wide as 25 mm. The dataset also includes images with a variety of obstructions, including shadows, surface roughness, scaling, edges, holes, and background debris. SDNET2018 can be useful for the continued development of concrete crack detection algorithms based on DL CNN, which are a subject of continued research in the field of structural health monitoring.

Another dataset was developed from Middle East Technical University (METU) building crack images (Özgenel , 2018). The dataset contains 40,000 image patches of cracks and non-cracks. The patches were prepared from 500 images of 4032 × 3024 pixels resolution acquired from walls and floors of several concrete buildings on the METU campus. Each full image was then separated into the crack and non-crack patches.

COncrete DEfect BRidge IMage (CODEBRIM) dataset was prepared and worked upon by Martin Mundt et. al. (Martin Mundt , 2019). The dataset comprises five common defect categories which are crack, spallation, exposed reinforcement bar, efflorescence (calcium leaching), and corrosion (stains), found in 30 unique bridges. The authors collected the dataset in challenging weather conditions to include wet and stained surfaces with multiple cameras at varying levels of scaling. A subset of the images was captured using a UAV to cover the large parts of bridges which are not easily accessible to humans. A total of 1590 high-resolution images were manually labelled in the defect classes. A multi-stage annotation process was adopted by the authors which involved the curation of acquired images, annotation of overlapping and non-overlapping bounding boxes per defect and sequentially labelling of each class separately. A total of 2507 cracks, 1898 spallation, 883 efflorescence, 1507 exposed bars, and 1559 corrosion stain images were prepared. Figure 31 shows the snapshot of CODEBRIM dataset available on the github link.

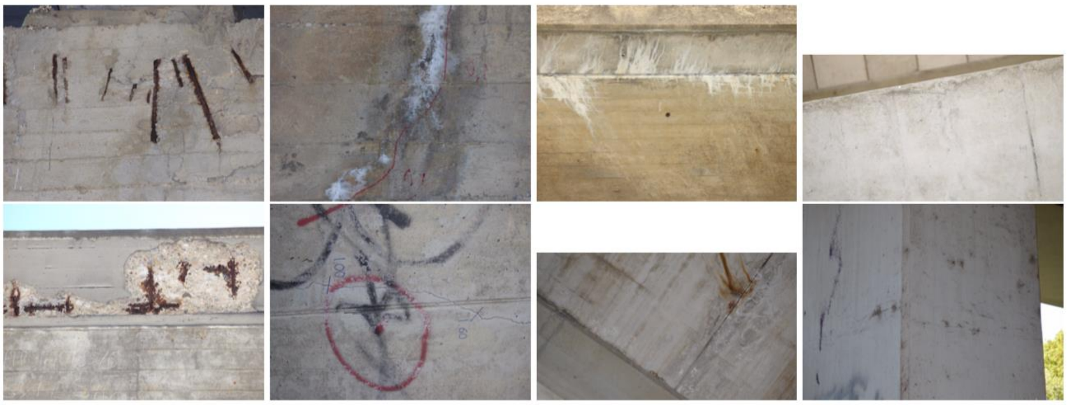


Figure 31 Snapshot of CODEBRIM dataset (meta-learning-CODEBRIM , 2022).

A dataset containing pavement surface cracks with pixel-wise annotation was proposed and used in the work presented by Qipei Mei et. al. (Qipei Mei , 2020). Another similar asphalt pavement labelled dataset was worked upon by Zhun Fan et. al. in their work on pavement crack detection using encoder-decoder architecture (Zhun Fan , 2020). Eberechi Ichi et. al. developed a Non-Destructive Evaluation (NDE) dataset named as Structural Defect NETwork 2021 (SDNET 2021) (Eberechi Ichi , 2022). The dataset consists of three NDE methods namely Infrared Thermography (IRT), Impact Echo (IE), and Ground Penetrating Radar (GPR) data collected from five in-service reinforced concrete bridge cracks. The prime objective of the dataset was to carry out concrete delamination detection and reinforcement corrosion detection.

Researchers who have addressed the crack detection application have created or generated their dataset with manual labelling (Changyon Kim , 2019). As observed in the dataset presented in the literature, the availability of crack-related datasets started in 2018 onwards. None of the previous datasets was prepared for standard concrete cubes used in laboratory testing. Due to the unavailability of the labeled dataset and ground truth images for laboratory-scale concrete cubes, the solution of the application becomes more challenging and time-consuming. However, an experimental data acquisition setup is developed in the present work. The setup is not only unable to image and load data acquisition but also performs real-time crack detection and monitoring during the compression test of the cube.

Normally crack detection using CNN is performed with two approaches. One is patch or contour-based and the other one is pixel based. In the contour-based approach, contours obtained after the segmentation operation are segregated into crack and no-crack classes manually. Each patch image either contains a crack or a non-crack. Both types of patches have some background surface of the cube. The second type of approach is pixel-based, each pixel in the image is annotated for cracks. Using an annotation tool or labeler tool, individual pixels are labelled into classes like cracks, non-cracks, and background depending on the objective and requirement of the application.

Dataset generation and labelling or ground truths are generated manually which is a time-consuming and tedious task. The contour-based approach is used in the present work. Initially, the binary contour dataset is prepared, trained and evaluated for performance which is discussed in detail in upcoming sections of this chapter. The results and observations are stated in the results sub-section of the chapter. Further, greyscale contour image data is prepared for training, testing and evaluation. Data imbalance is observed and to overcome that, data augmentation techniques are used. The balanced dataset, training and testing approach, and performance evaluation are discussed in the upcoming sections of the chapter.

4.5.4.1 Binary contour image dataset

Dataset collection and preparation for training, validation and testing is a critical step in the process of structuring a DL application. For the problem of crack and non-crack contour classification, a binary contour dataset is prepared using the images collected with the help of the experimental setup. A novel approach is adopted for the implementation of the binary classifier which can accurately classify cracks developed on the surface of laboratory-scale concrete cubes. The dataset required for the concrete cube crack classification is not available, due to which the challenging task is to prepare a sufficient dataset of the concrete crack images along with labels of crack and non-crack. To prepare the dataset, the images of concrete cube acquired using the developed experimental setup is used. Multiple concrete cubes are tested using the compression testing machine and image data is acquired during the test. Acquired images are pre-processed in order to remove the surface dents, noise, and shadow present on the concrete cube surface. After the binarization process using adaptive thresholding, contours are identified in the segmented image and extracted using the bounding box coordinates. These extracted contours mainly consist of crack and non-crack contours which are then manually separated into two classes namely crack class and non-crack class.

Figure 32 shows the flowchart of the dataset generation process. The raw greyscale images collected during the testing are then pre-processed and binarized. Contours are identified in the segmented result and extracted using the bounding box coordinates of each contour. These extracted contour images are saved in the memory and manually separated into two classes. The same process is carried out for more than 200 images of multiple concrete cubes captured during the compression test cycle. Each image contains around 70-230 numbers of contours depending on the cracks and noise, and dents present on the surface. A binary image would contain roughly between 0-20 crack contours and 100-250 non-crack contours. The number of cracks and non-crack contours largely depends on the pre-processing and threshold operation along with the cracks and non-cracks present on the surface of the concrete cube. Considering this fact, an optimum value of threshold was used such that all crack contours are properly segmented. A total of 18,296 numbers of non-crack contours and 3,468 numbers of crack contours are collected for the training, validation, and testing of the CNN.

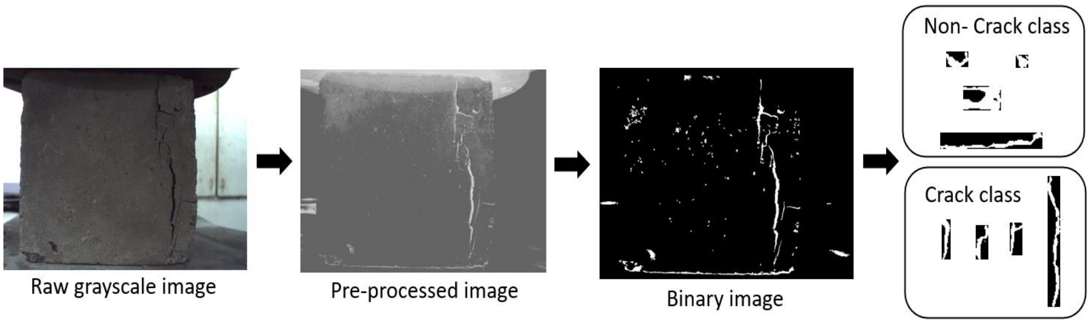


Figure 32. Flowchart of dataset generation using binary contour images.

Out of the collected dataset, 80% of the dataset is used for inception v3 training, 10% is used for validation and the remaining 10% is used for testing. Figure 33 shows a few samples of non-crack contour images out of the collected data while the samples of crack contour images are shown in Figure 34.

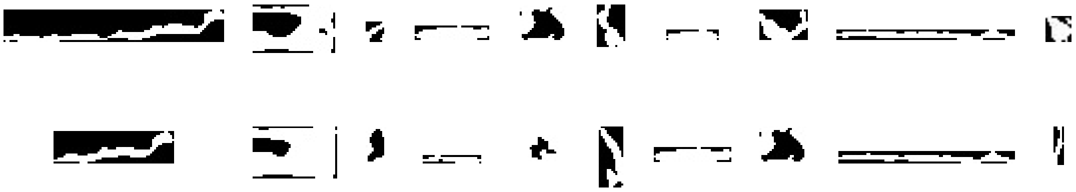


Figure 33. Sample images of non-crack binary contours.

Each of the non-crack contour images is displayed in two rows in Figure 33. Horizontal crack-like contours are the edges of the concrete cube segmented in the binarization process. Major noise, dents and edge-like shapes are observed in the non-crack contours. These contours look similar to small and broken cracks. Crack contour images are arranged in two rows as seen in Figure 34. Minor, small and large cracks are observed in the crack samples. The majority of the cracks are vertical, however, in a few cases can have branches as well which may extend in the diagonal direction.

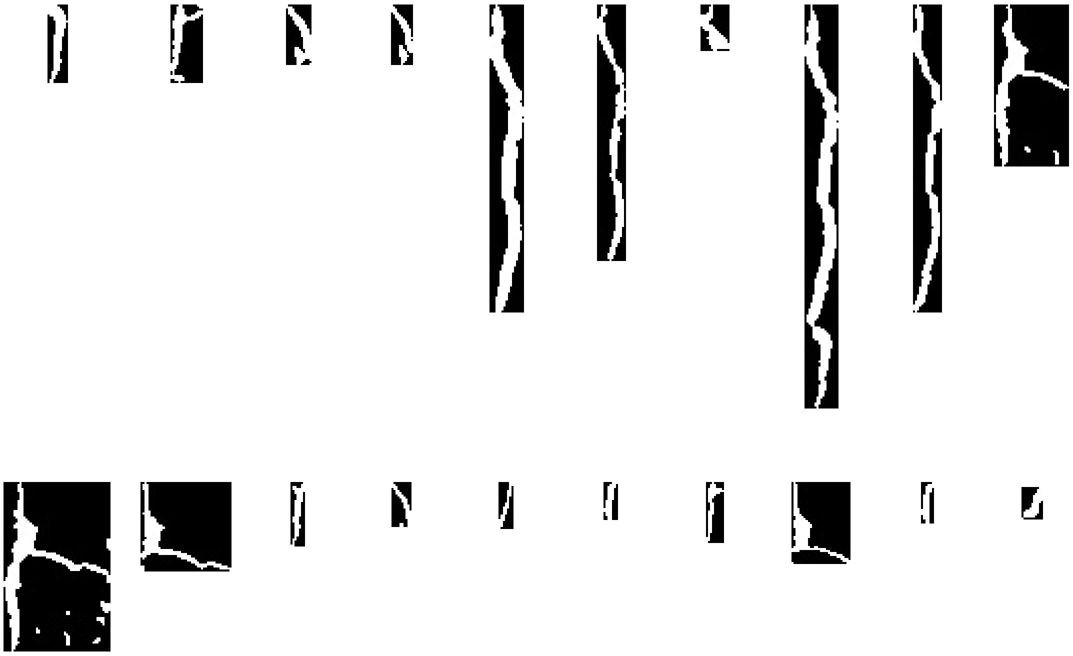


Figure 34. Sample images of crack binary contours.

From the samples of crack and non-crack binary contours, it is also evident that the shape of each contour image is different and inconsistent. From a shape point of view, a few non-crack contours may be disguised as crack contours. These contour images are utilized for the inception v3 model training, validation and testing. The model training and testing flowcharts are discussed in the next sub-section of the thesis. Greyscale dataset and data augmentation techniques are discussed further.

4.5.4.2 Greyscale contour image dataset

With reference to the process of dataset generation from binary segmented images, a similar process is carried out using greyscale images. Greyscale images contain much more information as compared to binary images. Binary contour images of cracks and non-cracks contain limited information related to the shape of the contour. This information may not be distinguishable in certain cases when the size and shape of crack contours match that of the non-crack contours. In order to overcome the issues, greyscale contour images of crack and non-crack are used for the model training. Figure 35 shows the dataset generation flowchart. The raw greyscale image collected during the compression test is pre-processed and binarized. Coordinates of the contours bounding box are identified and used to extract the contours from greyscale images. The coordinates identified from the binary images can be used to extract the contours from the greyscale image as there is no change in the spatial coordinates. These greyscale contour images are saved and manually labelled as cracks and non-cracks.

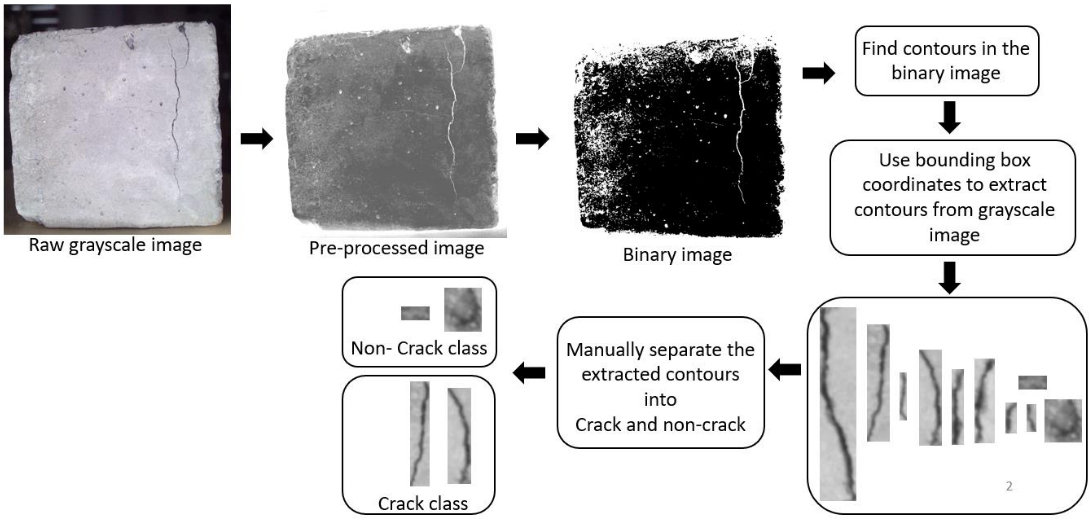

Raw grayscale image
Pre-processed image
Binary image
Find contours in the binary image
Use bounding box coordinates to extract contours from grayscale image
Manually separate the extracted contours into Crack and non-crack
Non- Crack class
Crack class
2

Figure 35. Flowchart of dataset generation using greyscale contour images.

The sequence of operations described in the flowchart is applied to 200 greyscale images of concrete cubes. Multiple concrete cubes are tested for compression and images are captured during the test cycle. Approximately 70-230 greyscale contours are found in each concrete cube image. The number of greyscale contours depends on the number of cracks and non-cracks present on the surface. A few samples of the non-crack contour images are displayed in Figure 36.

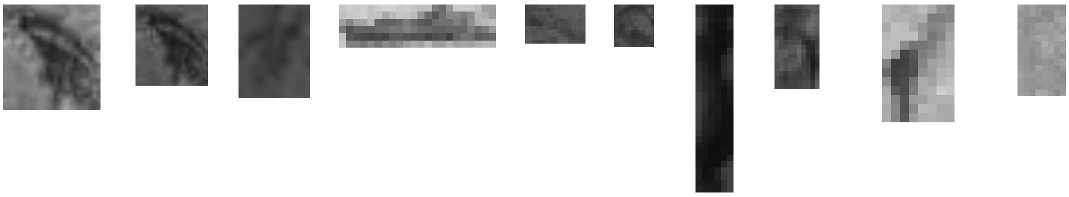


Figure 36. Sample images of non-crack greyscale contours.

A total of 3,468 greyscale crack contour images and 18,296 numbers of greyscale non-crack contour images are collected. These are manually separated into the crack and non-crack classes with utmost care and saved in the memory. The intensity variations in the non-crack contour images are evident from the samples shown in Figure 36. Few contour images are dark whereas others a bright, even though standard illumination and industrial-grade vision hardware are used to acquire the images. This is due to the inherent variations in the grey nature of the different concrete cubes.

Crack contour greyscale images are shown in Figure 37. Similar to non-crack contours, bright and dark crack contours are seen in the samples. Few contour images have single cracks whereas other images contain multiple or branched cracks. Out of which, the majority of cracks are thin in width and vertical in nature. Spalling of concrete also occurs during the compression test cycle and few crack contours showcase that.

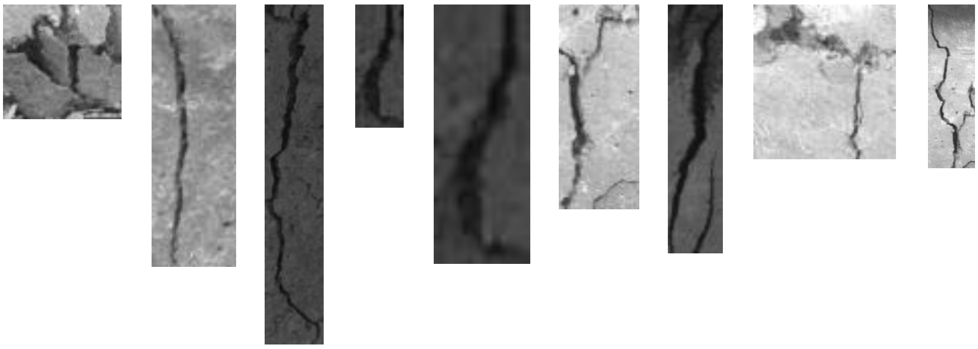
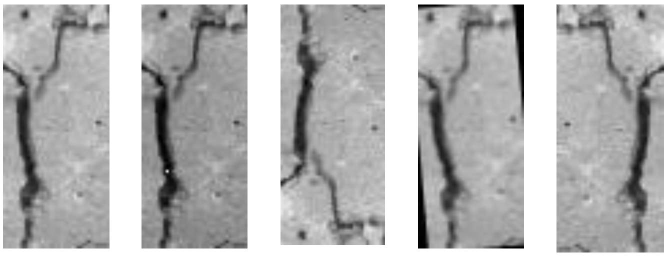


Figure 37. Sample images of crack greyscale contours.

In comparison with binary contours, greyscale contours have additional features considering intensity variations, texture features, and shape features. Significant variations are observed in the intensity levels of cracks which seems inconsistent. Looking into the number of cracks and non-crack contours images extracted, an imbalance in the class-wise dataset is apparent. As non-crack contours are always higher in numbers as compared to crack contours.

The details of model training, performance interpretation, evaluation and the results are discussed in the upcoming section of the chapter. In order to mitigate the issue of data imbalance, data augmentation techniques are used. The imbalance in the number of crack and non-crack contours gets reflected in the CNN model performance as well. Additionally, looking into the demand to provide sufficient variations in the dataset, the extracted crack and non-crack contour images are augmented using flip, rotation, and brightness adjustments operations.

In order to solve the issues, data augmentation techniques like vertical flip, horizontal flip, rotation, and brightness adjustments are applied on both crack and non-crack contours to enhance the dataset. In rotation, 5° counter-clockwise rotation is performed on the contour images while in brightness adjustment, images are first converted into HSV (H – Hue, S- Saturation, V-Value) colour space and intensity values 10 and 25 were added to the value. These augmentation techniques are applied to all the images of crack contours and only selected images of non-crack contours in order to balance the total dataset. Overall contour images per class are approximately 80,000 after the data augmentation. Exactly 79,445 greyscale contour images of cracks and 80,721 numbers of non-crack contour images are prepared for training. Figure 38 shows augmented images of a sample crack contour image. Sub-figure (a) is the sample crack contour image, (b) shows the brightness adjusted image, (c) shows the horizontal flip image, (d) shows the rotated image and € shows the vertical flip image.



(a) (b) € (d) €

Figure 38. Augmented images of a sample crack contour image.

The flowchart of the entire crack detection process is discussed in the crack the proposed methodology section. The process of contours extraction remains the same as the one used in the dataset generation. Each extracted greyscale contour is then passed through the trained model. The obtained predictions were used to highlight the respective contours with a green colour bounding box for cracks and a blue colour bounding box for non-cracks. Performance evaluation of the model for crack detection is discussed in the results and analysis section.

4.5.5 Model training and testing framework

Post baseline model selection and dataset preparation, training of selected model and measurement of its performance are performed. Interpretation of the model performance after training is an important step in a DL application. Many times, CNNs perform exceedingly well and yet it is difficult to understand why they worked well (Elgendy , 2020). Due to this, CNNs are often criticized and called ‘black boxes’. Accurate crack detection is performed with the help of Inception v3 which is a popular and widely used CNN model. The DL network has shown a low error rate with fewer parameters and used the concept of factorization wherein the convolution operations are factorized to reduce the number of connections and parameters without decreasing the model efficiency. The inception v3 model is capable of handling input images of variable size and provides an efficient classification. The architecture of Inceptionv3 is shown in Figure 39.

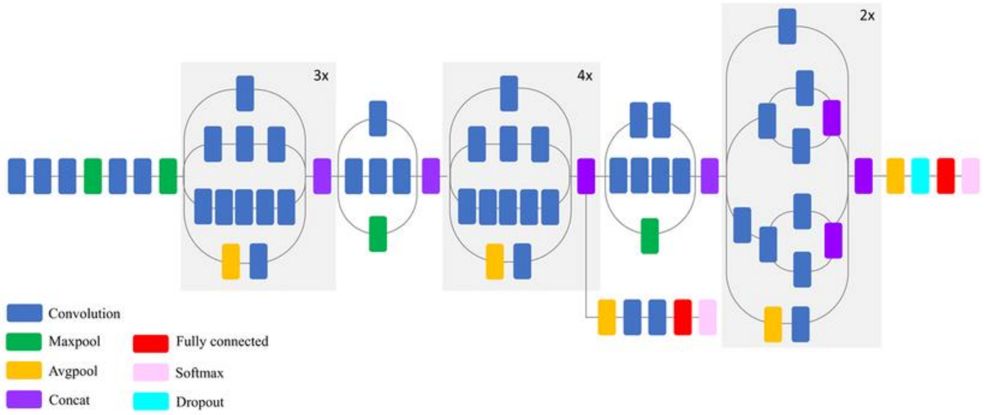


Figure 39 Architecture of Inceptionv3.

The unique features of the implemented methodology are the development of the experimental setup, data set collection and preparation of the concrete cube dataset, and the novel framework of crack detection in concrete cubes using CNN model. The dataset generation flow and crack detection flow are developed to improve the overall accuracy of crack detection. The flowcharts of model training and testing in the case of a binary dataset, imbalanced greyscale dataset and augmented greyscale dataset is discussed further with necessary flow diagrams.

Determination of any bottlenecks and diagnosis of the CNN model components which are performing poorly is carried out after the training. If the model gives poor results, identification of overfitting, underfitting or data insufficiency is equally significant. One of the main causes of poor performance in DL is either underfitting or overfitting the dataset. Instead of checking the verbose output and comparing the numbers, plotting the curves throughout the training can result in a better diagnosis of overfitting and underfitting. Figure 40 depicts the flowchart of the training inception v3 CNN model using the prepared dataset. The training process till the extraction of binary contour images remains similar to the one used in the dataset generation.

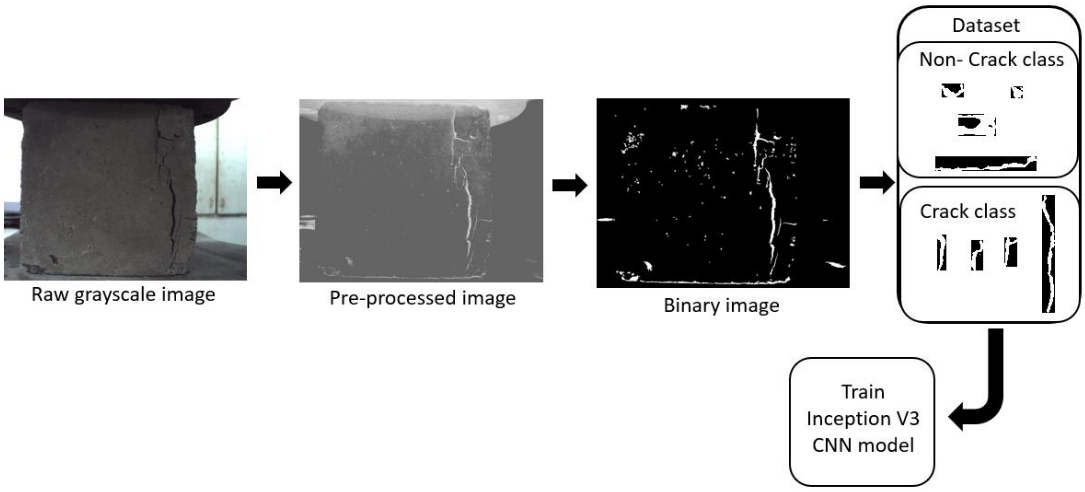


Figure 40. Flowchart of CNN model training using binary image dataset.

The collected dataset is fed for model training. The testing of the trained model is carried out using the flowchart shown in Figure 41.

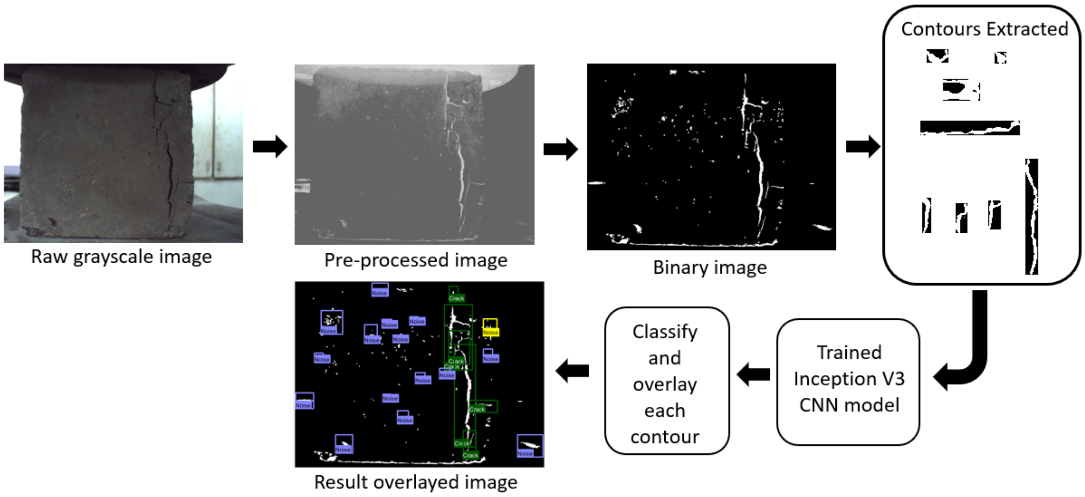


Figure 41. Flowchart of CNN model testing using binary image dataset.

The raw greyscale images which are not used for training and validation are processed to extract crack and non-crack contour sub-images. These images are transferred to the re-trained inception v3 model for the prediction of crack and non-crack class. The predictions are overlayed on the raw greyscale image using a bounding box. Predicted crack contours are shown with a green colour bounding box whereas non-crack contours are shown with a blue colour bounding box. The performance analysis of the testing is covered in the results and discussion section of the chapter.

Figure 42 depicts the flowchart of the data set generated from the acquired images and model training. Raw greyscale images acquired with the image acquisition setup are pre-processed initially and converted into a binary image using a thresholding operation. Multiple contours are extracted from the greyscale image by using the bounding box coordinates of contours found in the binary image. The prepared dataset is used for the training, validation and testing of the inception v3 model.

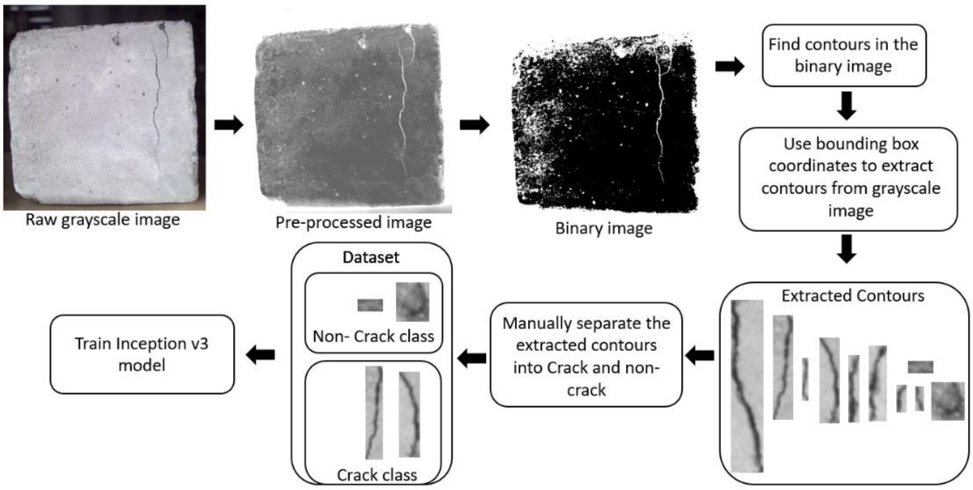

Raw grayscale image
Pre-processed image
Binary image
Find contours in the binary image
Use bounding box coordinates to extract contours from grayscale image
Extracted Contours
Manually separate the extracted contours into Crack and non-crack
Dataset
Non- Crack class
Crack class
Train Inception v3 model

Figure 42. Flowchart of CNN model training using greyscale image dataset.

The figures show the combined approach of the contour extraction process, classification and visualization. The raw greyscale images are processed to extract crack and non-crack contour sub-images. These images are transferred to the re-trained inception v3 model for the prediction of crack and non-crack class. The predictions are overlayed on the raw greyscale image using the bounding box. Predicted crack contours are shown with a green colour bounding box whereas non-crack contours are shown with a blue colour bounding box.

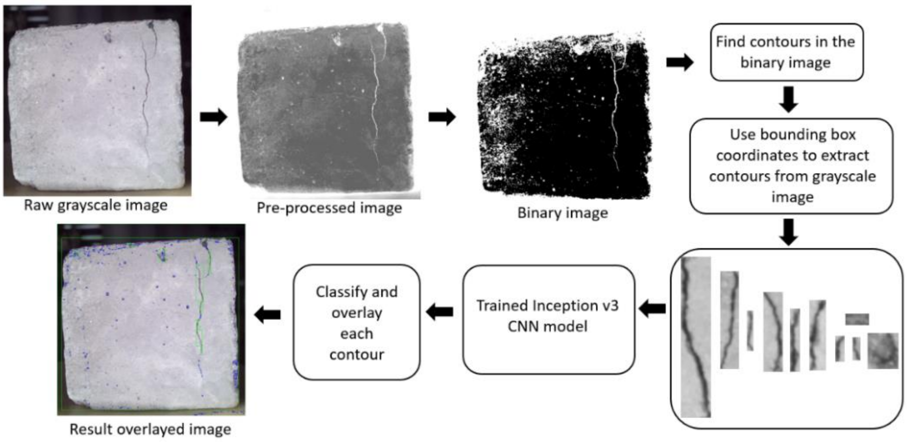

Raw grayscale image
Pre-processed image
Binary image
Find contours in the binary image
Use bounding box coordinates to extract contours from grayscale image
Trained Inception v3 CNN model
Classify and overlay each contour
Result overlayed image

Figure 43. Flowchart of CNN model testing on greyscale images dataset.

The model training and testing flowchart shown in Figure 42 and Figure 43 are used for both the imbalanced greyscale dataset as well as the augmented greyscale dataset. However, the model training related details like parameters, hyperparameters, training results and observations are discussed for only balanced and augmented greyscale datasets.

The generated dataset was divided into a train, validate and test split. 80% of the data is used for training, 10% of the is used for validation and the remaining 10% of the data is used for the model testing. This testing is carried out to evaluate the final accuracy of the model after the training is complete. All the possible variations in both crack and non-crack classes should be fed for the training to achieve an optimum CNN model performance. For validation, contours which are not used for training or testing are used. For training, validation and testing of the inception v3 model, a python code is implemented. Tensorflow, Numpy, and various other libraries are imported for the training of the model. The trained model is saved as .pb file and the labels are stored in a text file. The .pb format is the protocol buffer format used in Tensorflow to hold models. The format is a general way to store data by Google because it is much nicer to transport, as it compacts the data more efficiently and enforces a structure to the data. The number of epochs is kept at 5,000 while the initial learning rate is kept at 0.01. The batch size is kept at 100 and the number of batches is 1,600 for the training.

The number of epochs is initially kept lower and then gradually increased to obtain a satisfactory training and validation performance. The initial learning rate of 0.01 for training is standard and used in the present work as well. The learning rate is kept adaptive to the learning due to which optimum performance in training can be obtained. The batch size is chosen as 100 and the number of batches is chosen as 1600 because the total dataset size is 160000 sub-images of crack and non-crack. These sub-images are smaller in size and the amount of data is much lesser in comparison to a full image. Hence, a large batch size of 100 sub-images can be fed at a time to the model for training. During the training process, the validation is carried out using the entire validation set to achieve more stable results across the training iterations. Once the training is complete after the specified number of epochs, the final accuracy testing of the model is carried out using the entire training set. This testing set is used only once after the training is completed.

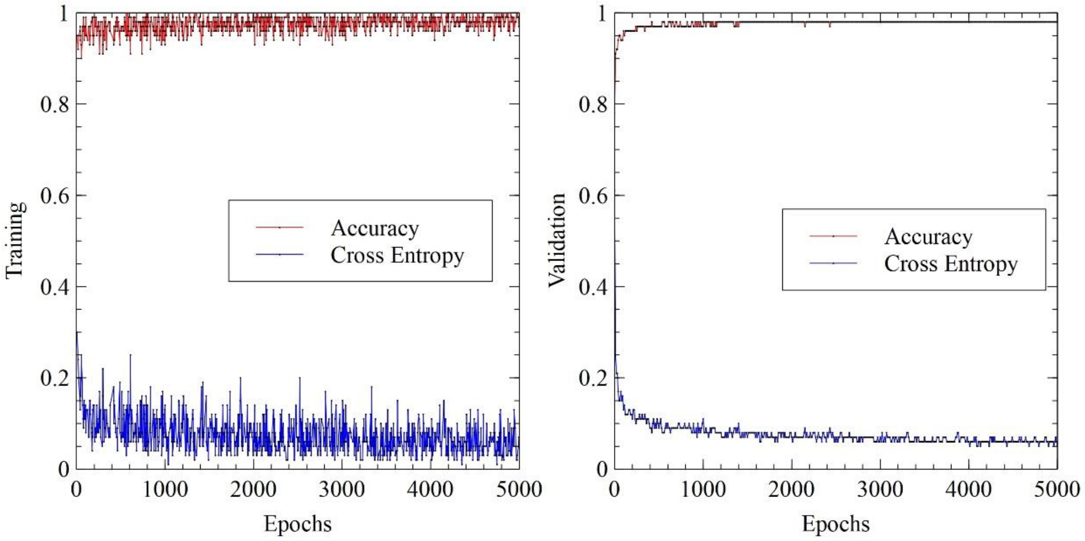


Figure 44. Training and validation performance of inception v3.

The model has a 1000 number of layers in the output layer because inception v3 was used for the classification of the image from ImageNet database. The present application is focused on the binary classification of crack and non-crack contours. And hence, the model architecture is modified in order to meet the requirements. The final dense layer in the model is updated with two neurons in order to classify cracks and non-cracks. The softmax function is used as the activation function in the output layer and cross-entropy is used as the loss function to train the model. The gradient descent optimizer available in Tensorflow is used for the optimization of mean cross entropy which is the loss function used. Other essential hyperparameters and model requirements are defined as per the standard or kept unchanged. Figure 44 shows the graph of the epochs versus accuracy and cross-entropy of training and validation of the model.

An average of 97.49% accuracy and 7.38% cross-entropy are achieved in the training whereas 97.67% accuracy and 7.69% cross-entropy are achieved in the model validation. The training and validation performance of the inception v3 model trained with a balanced and augmented greyscale contour image dataset is presented. And the best model performance is shown in Figure 44. Both the training and validation data shows variability in terms of accuracy and cross entropy plotted with respect to epochs. Similar variations are observed in various other applications. One of the primary reasons of the observation due to that fact that the model is going through the backpropagation while training on the dataset and the model weights are being adjusted here with each epoch to minimize the cross entropy. While in case of the validation, the model weights are not updated, just error and cross-entropy is calculated after every training epoch.

Various other instances of model training, evaluation and interpretation were carried out before this. Various other instances of model training, evaluation and interpretation were carried out before this. These instances are carried out in order to reach to the optimal hyperparameters for best performance of the model. In addition to that, the dataset is updated multiple times and hence the model is also trained after each update. The initial training instances were carried out on the Google Colab platform. However, later on, a personal computer was used. The configurations of the system are described in the hardware selection sub-section of chapter 3. The model is trained properly and no signs of underfitting or overfitting are observed, which is evident from the graphs shown in Figure 44. Results in form of parametric assessment, observations and crack detection result images are discussed in the results and discussion sub-section of the chapter.

4.5.6 Real-time implementation approach

For software development, LabVIEW and Python are used. As discussed in the software selection stated in section 3.3 of the thesis, image acquisition, load acquisition, image processing, classification, results display, crack-load monitoring, data logging and other essential tasks are performed using the developed software. Out of the mentioned tasks to be performed by the software, only the classification task is carried out using Python and Tensorflow. As NI LabVIEW does not have a good and open-source library and toolkit required CNN development, Python along with Tensorflow and other required libraries are used. Inception v3 model training, validation, testing and performance evaluation iterations are carried out using Python. In addition to that, a ML -based approach is also developed using Python. Other crack detections and pre-processing methods are implemented using both Python and LabVIEW.

Looking into the objective of crack-load monitoring and analysis along with real-time crack detection, the software should be able to carry out all the necessary tasks in real time during the compression test of a concrete cube. On average, the compression test of a concrete cube takes approximately 90 – 350 seconds. The observation is with respect to the number of tests conducted during the work. More than 200 test were conducted on concrete cubes using the same compression testing machine and loading rate. The amount of time may vary based on the loading rate, strength of concrete cube and other factors. The time may be lower or higher depending on the loading rate and the strength of the concrete cube subjected to the compression test. In order to achieve real-time crack detection and analysis, a processing rate of 1 sample of image per second should be achieved. All the tasks mentioned above should be completed between 1,000 milliseconds.

Initially, the computation time performance is checked using the personal computer. The computer has AMD Ryzen 5 3500 U processor, the base clock frequency is 2.10 GHz, the maximum boost clock frequency is 3.7 GHz, 4 cores, 8 threads, 12 GB physical RAM, 512 GB SSD, and Windows 10 home version. Figure 45 shows the crack detection flow diagram and each step is marked with the software platform used. The crack detection is performed on the computer. The development of crack detection classification using the inception v3 model is carried out using Python and Tensorflow.

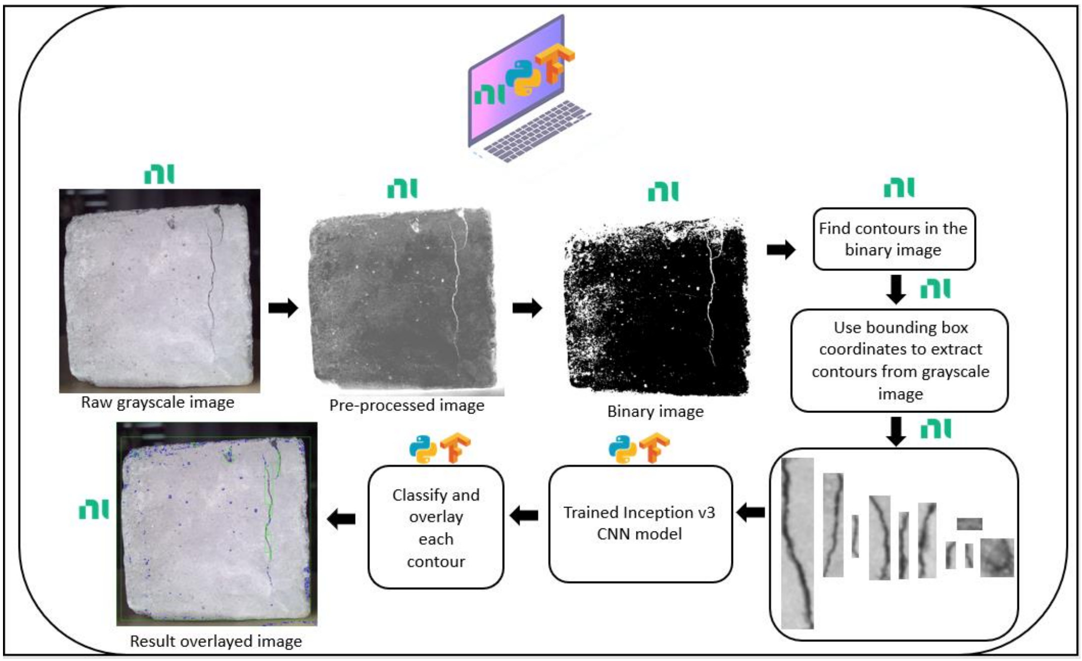

Raw grayscale image
Pre-processed image
Binary image
Find contours in the binary image
Use bounding box coordinates to extract contours from grayscale image
Trained Inception v3 CNN model
Classify and overlay each contour
Result overlayed image

Figure 45. Crack detection performed on CPU using LabVIEW and Python.

The acquisition of concrete cube images subjected to compression test is performed using LabVIEW along with load data acquisition. Post-acquisition, pre-processing, segmentation, contour identification and contour extraction are performed using LabVIEW. Each contour is saved in the computer memory at a fixed drive location and with a known name. A python function is called by LabVIEW in order to obtain the model prediction for the saved contour image. The save images are called in the Python function and with the use of a trained model file, a classification is obtained. This classification is returned by the Python function to LabVIEW. The result is overlayed in the greyscale image to identify the crack and non-cracks predicted by the CNN model. As mentioned earlier, the software should complete all the tasks in 1,000 milliseconds in order to achieve a rate of 1 image per second. The processing capacity of the CPU available in the computer is not enough to achieve the required performance. The analysis of computation time performance is discussed in the results section of the chapter.

In order to fulfil the computation demand of real-time crack detection, a graphical Processing Unit (GPU) is suitable. However, due to the unavailability of a GPU resource, an alternative processing system is used. Addressing the demands of low-cost and add-on computation resources specifically for AI and ML applications, USB accelerators are developed by Google and Intel. A USB accelerator from the Google Coral family and a second-generation neural compute stick is available from Intel (USB Accelerator , 2022) (Intel® Neural Compute Stick 2 (Intel® NCS2) , 2022). These USB accelerators provide add-on computing power required for the implementation of AI and ML applications on low-configuration computation devices like Raspberry Pi and other similar single-board computers. These accelerators can be easily used with personal computers as well. The cost of both these devices is under INR 10,000/-, which is comparatively lower than the cost of a standard GPU required for AI and ML applications.

Google coral USB accelerator is used for the purpose of improving the computation time of the inception v3 classifier. The USB accelerator is also called an edge Tensor Processing Unit (TPU). TPU is an AI accelerator Application-Specific Integrated Circuit (ASIC) developed specifically for neural network ML in the TensorFlow ML library. The edge TPU provides an addon coprocessor to the existing computation system. It enables the system to perform high-speed inferencing of ML and DL models. It is connected to USB3.0 of the computer and supports Windows, Linus and macOS operating systems. The edge TPU has the capacity to perform 4 Trillion Operations Per Second (TOPS) and only uses 0.5-watt power for each TOPS. In the performance comparison of the device, it is stated inference time of Inception v1 architecture with input image size 224 × 224 is 90 milliseconds on the desktop CPU whereas the same model when tested on the desktop CPU alongside the USB accelerator gets executed in 3.4 milliseconds. The execution on CPU + TPU is approximately 26 times faster than the execution on CPU only as discussed in Table 11 in the sub topic 4.6.6 results of real-time implementation. Similarly, inference times of other popular models like Unet, DeepLab v3, DenseNet, MobileNet v1, ResNet v1, VGG16, VGG19, EfficientNet and others are officially available for comparison on the website (Edge TPU performance benchmarks , 2022).

Looking into the comparison of inference time and improvement in the execution time, the TPU device is used for the classification of crack and non-crack contours. However, the model trained on a CPU has to be converted into a TPU-compatible model using a Tensorflow lite converter. The device only supports Tensorflow lite models which are fully 8-bit quantized and compiled specifically for the edge TPU (TensorFlow models on the Edge TPU , 2022). Figure 46 shows the crack detection flow diagram wherein each step is marked with the software platform used. In contrast to the flow diagram depicted in Figure 45, the classification of the contour is performed on TPU using Python and Tensorflow.

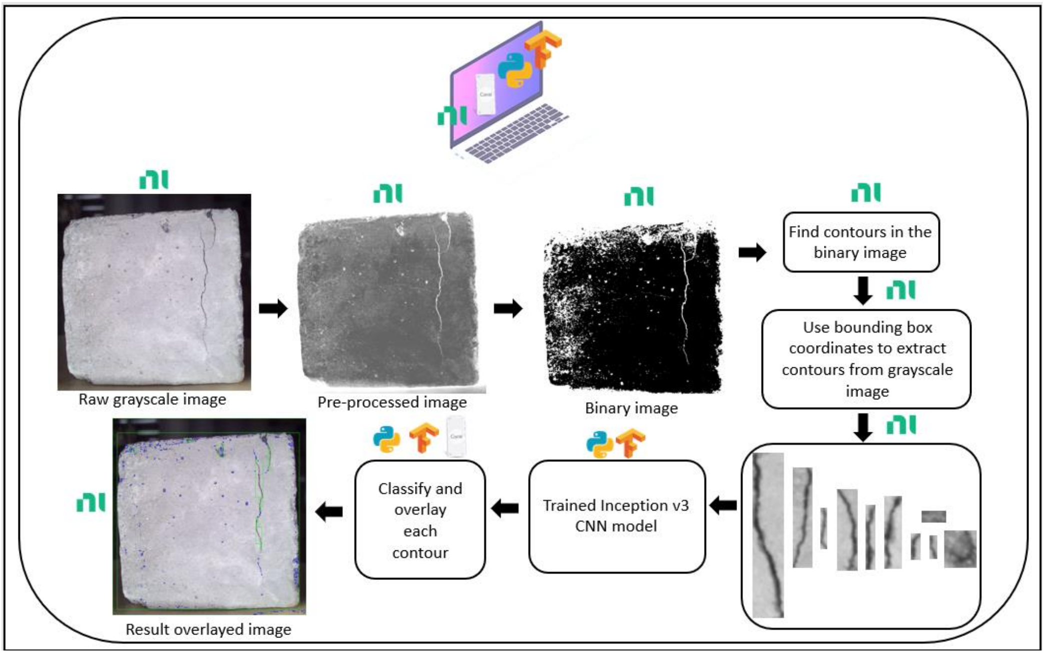

Raw grayscale image
Pre-processed image
Binary image
Find contours in the binary image
Use bounding box coordinates to extract contours from grayscale image
Trained Inception v3 CNN model
Classify and overlay each contour
Result overlayed image

Figure 46. Crack detection performed on CPU + TPU using LabVIEW and Python.

In order to improve the computation of the inception v3 CNN model trained to classify cracks and non-cracks, a USB accelerator from Google Coral is used alongside the CPU processing available in the computer. The computation time performance and comparison with the performance of crack detection using only CPU and CPU + TPU are discussed in the results section of the chapter.

4.6 Results and discussions

Efficient crack detection is very vital in the overall structural health monitoring of the infrastructures. The performance and efficiency of each processing method along with the current age CNN are analysed in the sub-section of the chapter. The quantitative data indicating the performance of each of the methods along with result images are presented for each of the methods in the results and discussions section.

4.6.1 Image pre-processing

Pre-processing operations are an essential step in the majority of image-related applications as they serve to improve the image for further segmentation and detection operations. Given the importance of pre-processing operations, filter, subtraction, and contrast stretching operations are chosen to improve the images of concrete cube surfaces. Result images of each step along with ROC analysis and computation time performance are presented to justify the use of these operations.

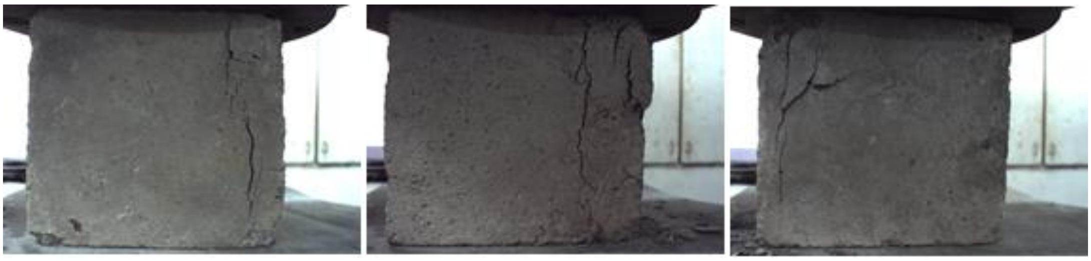


Figure 47. Sample images of concrete cube surface.

Figure 47 shows sample concrete crack images considered for the study of pre-processing operations. Figure 48 shows the result of the bilateral filter operation applied to sample images of Figure 47. The result indicates that the edges were preserved and noises were removed. Figure 49 illustrates the resulting image of the subtraction pre-processing method applied to sample images.

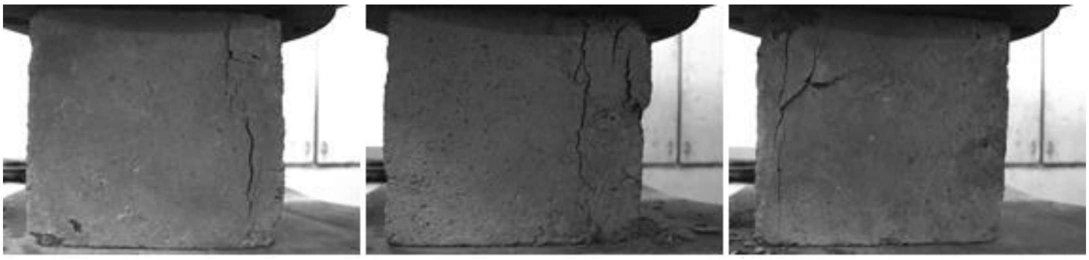


Figure 48. Result images after bilateral filter operation.

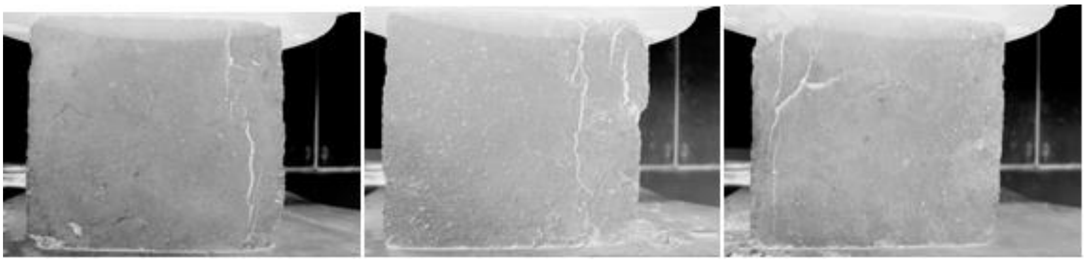


Figure 49. Result images after subtraction operation.

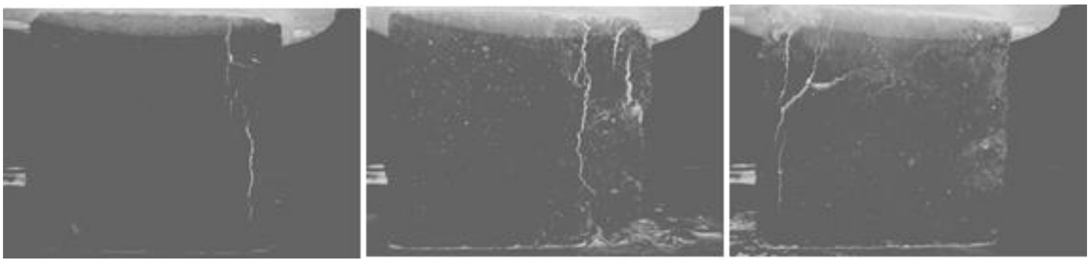


Figure 50. Result images after contrast stretching operation.

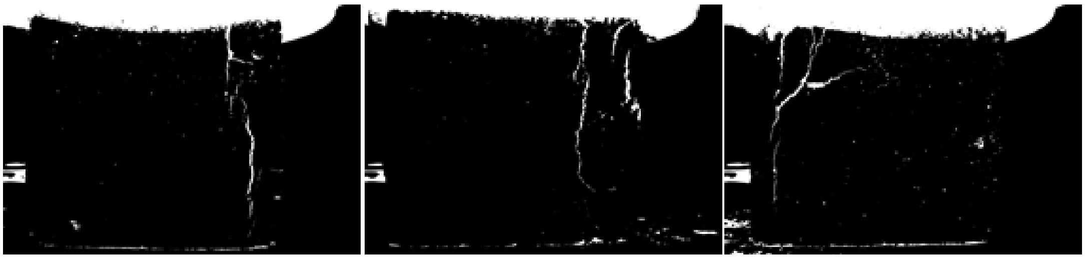


Figure 51. Result images after local threshold operation.

Figure 50 shows the image obtained after applying contrast stretching on subtracted images applied to sample images. The result indicates shaded parts and noise which are considered cracks due to the similar intensity values. To perform accurate crack detection, Niblack local thresholding is used and the result can be seen in the images shown in Figure 51.

The pre-processing methodology is tested for crack detection on more than 50 images similar to the images seen in Figure 47. All sample images are acquired using the developed experimental setup. The crack detection result can be visually observed from the result images shown in Figure 51. However, in order to evaluate the performance of the methodology, the receiver operating characteristics analysis presented in Table 3 is carried out.

Table 3. ROC analysis.

No. of crack pixels in ideal	**No. of crack pixels in processed**	**True positive fraction (TPF)**	**No. of background pixels in ideal image**	**No. of background pixels in processed**	**False positive fraction (FPF)**
21806	18175	83.34	3123922	3127553	16.65
28697	25667	89.44	3117031	3120061	10.55
19863	16101	81.06	3125865	3129627	18.93
20303	18130	89.29	3125425	3127598	10.70
9766	8200	83.96	3135962	3137528	16.03
5738	4744	82.67	3139990	3140984	17.33
7543	5945	78.81	3138185	3139783	21.19

The image size is 2048 × 1536 and a total of 3145728 pixels. Crack detection results of the chosen sample images were visually better in terms of crack detection. True Positive Fraction (TPF) is the ratio of the number of crack pixels in the processed image to the number of crack pixels in an ideal image. It signifies the ability of the method to be able correctly to identify cracks in the image. The method should also be able to identify background as the background which is evaluated as a False Positive Fraction (FPF). It is the ratio of the number of background pixels in the processed image to the number of background pixels in the ideal image. TPF should be near 1 and FPF should be near 0 which is observed in Table 3.

National Instruments LabVIEW vision assistant tool is utilized to obtain the number of crack pixels and the number of background pixels in both ideal and processed images. The ideal number of crack pixels are calculated manually for each of the crack in the image. The data mentioned in Table 3 were obtained using the area parameter of the particle analysis function.

Further testing of the present pre-processing methodology is carried out on a different set of images which consists of various types of surfaces. The majority of the cracks are identified easily if there is a wide difference in the contrast values of crack parts and background. Figure 52 sample image number 1 is acquired from the white wall of a building; image number 2 is a sample image of the crack surface captured from the corridor which consists slightly uneven surface and image number 3 is a sample image taken from a road which has high unevenness and the low difference between crack and non-crack areas.

Figure 53 shows the local threshold result of each test image shown in Figure 52. It is seen that a combination of subtraction pre-processing and contrast stretching method is effective to get more accurate crack detection results on the majority of the surfaces. The method can be very effective to find a crack if there is a significant difference in intensity value between foreground and background but for images with low contrast different like road surface shown in image number 3 of Figure 52, crack detection is poor.

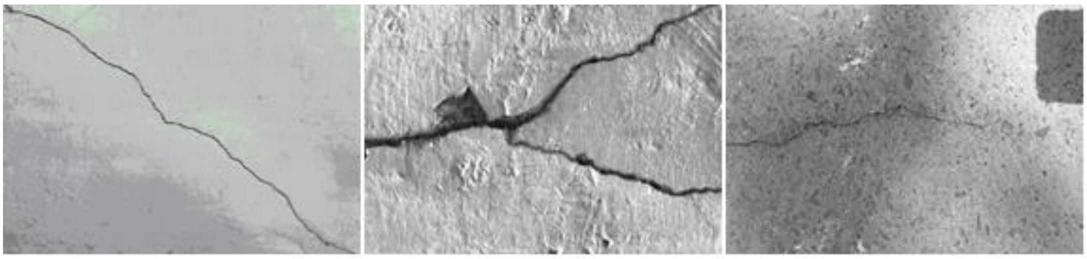


Figure 52. Testing images of surfaces with crack.

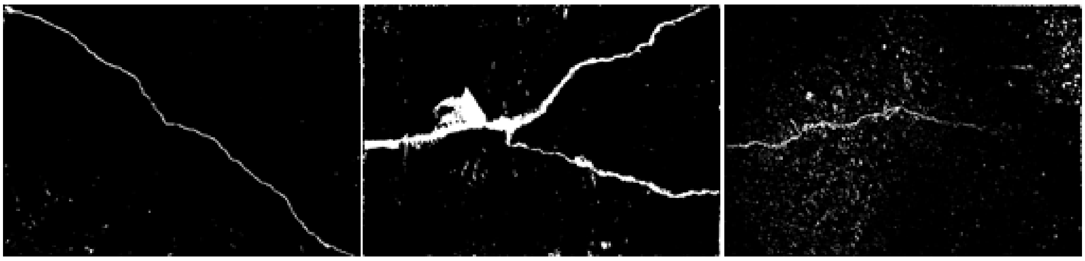


Figure 53. Result of pre-processing applied on testing images.

The present image pre-processing methods consist of two pre-processing operations and two processing operations which are implemented to improve the crack detection on different images on various crack surfaces. The methods are found effective when it is tested on images with distinguishable intensity difference between cracks area and background. Random unevenness on the surface of concrete cubes presents challenges in the development of robust image processing algorithms for accurate crack detection. The proposed method detects cracks on the surface accurately. Also, it is evident from the results that, a crack detection algorithm designed to identify cracks on the surface of a bridge surface will not provide equivalent accurate results when it is evaluated for different surfaces of concrete containing cracks. The developed algorithm gives accurate results on the majority of the crack containing concrete surfaces as seen in Figure 51, Figure 53 and the ROC analysis presented in Table 3.

4.6.2 Conventional image processing

Traditionally image processing methods and operations are used in a developed sequence with optimized parameters in order to achieve the objective of a vision application. Image segmentation is one of the widely used methods in a vision application because it separated the object of interest from the background based on the similarity in intensity levels or discontinuity in the intensity levels. Crack detection is a segmentation problem focussed to identify the cracks in the surface images of infrastructures. Multiple methods have been developed in order to detect the cracks in surface images of standard concrete cubes of size 150 mm × 150 mm × 150 mm which are used in the present work. Result images of each of these methods are discussed along with observations and reported issues.

4.6.2.1 Method 1 - Subtraction, line emphasis and iterative threshold

The crack detection method comprises two pre-processing operations and thresholding for segmentation. A median filter is convoluted over the original image to obtain a smoothed image which is then subtracted from the original image. The smoothing operation is used to remove thin line structures associated with cracks and the subtraction will remove the minor variations in the intensity levels. In order to obtain an optimum threshold value, each threshold value from the range 0 to 255 is used to segment the image. Each of the segmentation images is then evaluated using TPF and FPF. A similar process is carried on with multiple such concrete cube images in order to obtain the optimum threshold value and accurate crack detection results.

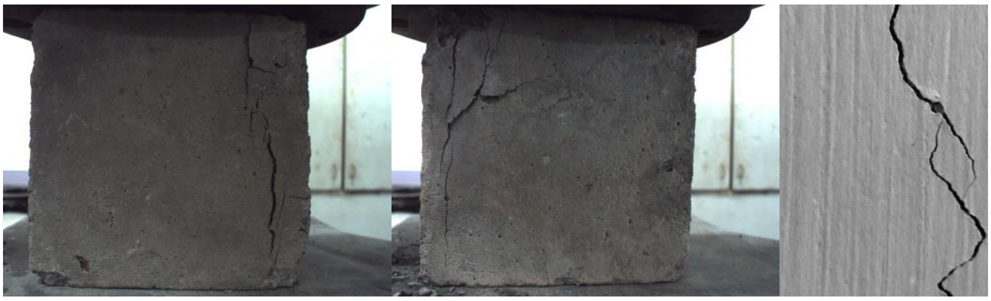


Figure 54. Testing images of cracks for line emphasis method.

Figure 54 shows the three samples of concrete surfaces with cracks used for the testing of method 1 described in sub section 4.3.1. Two out of the three sample images are of the concrete cube under compression test while the third image is of a building wall containing a crack. Figure 55. shows the results of the three samples of concrete surfaces with cracks used for the testing of method 1 described in sub section 4.3.1.

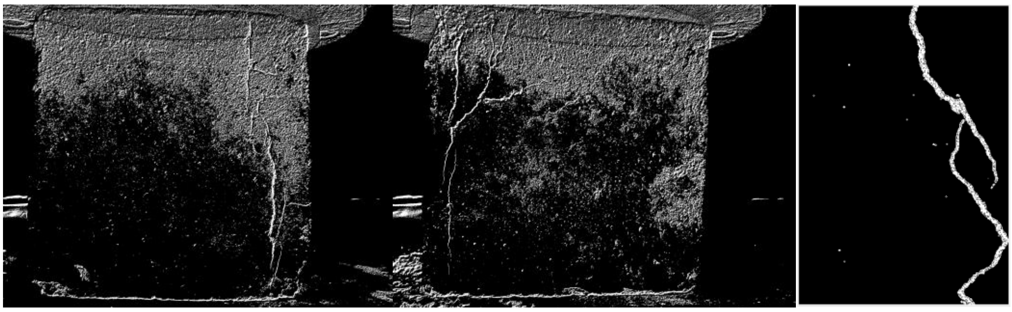


Figure 55. Result images of line emphasis method.

Although the cracks are highlighted in the result in image number 1 and 2, a lot of noise and dents are also detected which is visually evident. This issue can be due to the presence of unwanted noise and dents in the concrete cube image and the similarity in the intensity levels of dents and cracks. However, in the case of the third sample image wherein the background is homogeneous and the crack is dark with no presence of noise, dents or other unwanted non-crack parts, the crack segmentation results are accurate. The cracks can be seen in the segmented result image. The line emphasis-based method is used on the principle of amplifying the difference between the crack and non-crack elements of the image by emphasizing the edge and line-like structure. But not all the images have proper conditioning where the edges of the cracks can be highlighted due to hindrances like illumination conditions, and background and foreground intensity differences. Also, the dents, noises and other similar cracks-like structures present over the inspected surface fulfil the conditions of the eigenvalues partially. These issues affect the overall crack detection accuracy to a greater extent and hinder its use. It can be concluded from the result images that the methodology is suitable for crack detection in cases with no interfering areas, especially noises and dents.

4.6.2.2 Method 2 – Percolation processing

The method of the percolation model is analogous to the occurrence of the crack on the surface of a concrete cube and the propagation of cracks (Tomoyuki Yamaguchi , 2008). Looking into the results and observations claimed by the authors Tomoyuki Yamaguchi et. al. in their work, the percolation method is applied for obtaining accurate crack detection in surface images of concrete cubes under a compression test cycle. Since the percolation algorithm starts with a seed point which is essentially an edge point or the starting point of a crack, the methodology is applied and tested on small sub-parts of concrete cube images. In addition to that, the seed points are assumed to be the starting points of a crack. The images shown in Figure 56 shows the images of a concrete cube and a cracked wall of a building. These images are used for the testing the percolation based crack detection method discussed in the sub section 4.3.2.

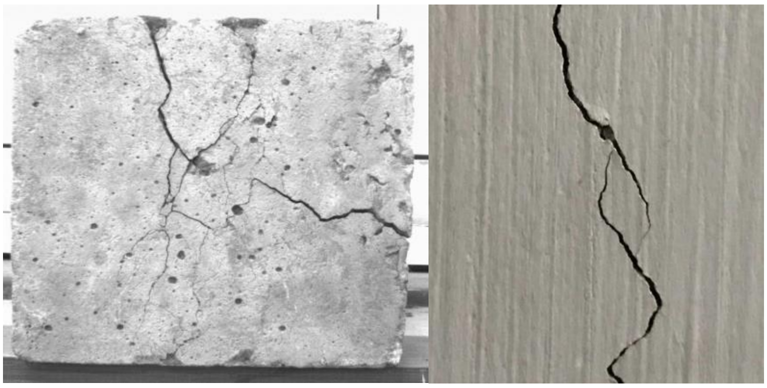


Figure 56. Testing images of cracks for percolation method.

In the percolation method, the brighter pixels in the local lattice are percolated and the darker ones are disregarded. With the use of pre-processing and processing operations, the problems of noise, dents and blemishes tend to reduce and even be completely removed in a few cases. The result images are shown in Figure 57. It is visually evident that the crack detection is reasonably good in the case of the concrete cube image while it is showing high accuracy in the case of the wall image. However, closely observing the result and original image of a concrete cube in Figure 56 and Figure 57, it is evident that minor and micro-cracks are not percolated. On the other side, a few dents and noises areas of the cube are percolated. In addition to that, longer cracks are not detected completely i.e., certain cracks appear broken in the segmentation result while they are connected in the original image.

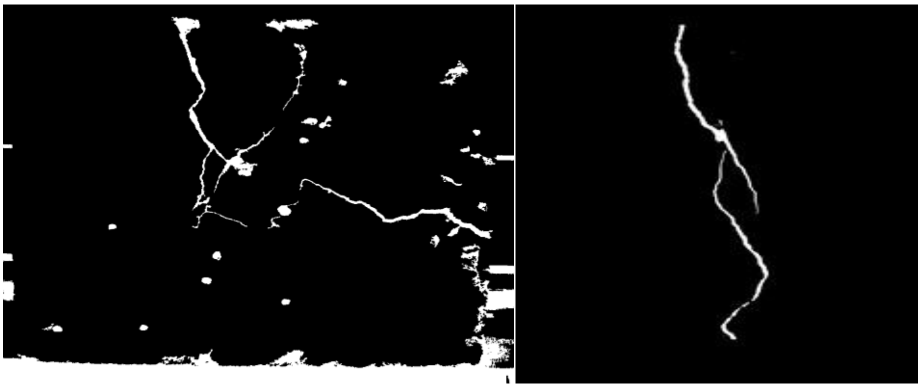


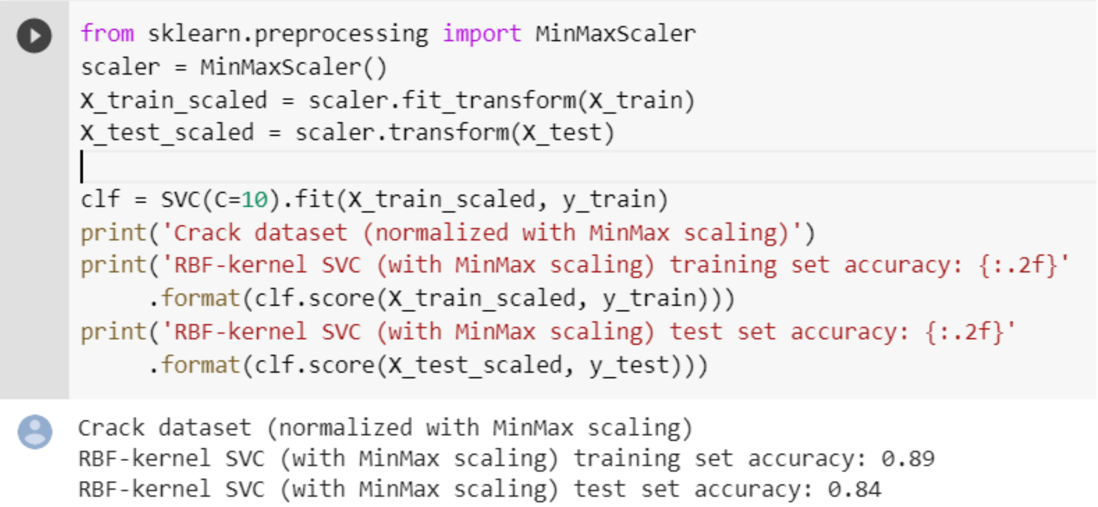
Figure 57. Result images of percolation method.

In a conclusion, the results are similar to the ones obtained with method 1 comprising subtraction, line emphasis and iterative threshold. Both methods 1 and 2 are able to detect cracks accurately in the building wall image, while in the case of the concrete image the crack detection results are not encouraging. Hence, the performance analysis of both methods is not evaluated. The percolation-based crack detection in concrete cubes is also affected by noises and dents inherently present on the surface. In addition to that, the selection of a seed point is also critical for the algorithm along with the configuration parameters like window size, threshold and others. Furthermore, percolation being an iterative method is a highly computationally demanding algorithm. Due to the mentioned issues and limitations, the algorithm is not suitable for crack detection in concrete cube images.

4.6.3 Machine learning

As stated in the results discussion of pre-processing, line emphasis and percolation methods, due to similarity in the intensities of cracks and non-crack parts, both get segmented in these methods. These inaccuracies immensely affect the overall objective of crack detection and crack monitoring. In order to monitor the cracks, properties of cracks like length, width and area will be calculated and monitored with respect to the applied load. If non-crack parts appear in the segmented results along with cracks, the calculation of crack properties will be completely wrong. Hence, the problem of crack detection is further updated into the classification problem where in the objective of the algorithm is to classify the segmented regions into crack and non-crack.

Referring to the ML domain, the problem of identifying a region as crack or non-crack can be categorised into a binary classification problem. KNN and SVM algorithms are used for the classification. Data processing is carried out using pandas while the sklearn library is used for the models. Programming, training and testing are carried out on Google Colab. Out of various region properties, bounding rectangle diagonal, perimeter, maximum ferret diameter, equivalent rectangle diagonal, area, elongation factor, compactness factor and Heywood circularity factor are selected and extracted using particle analysis in NI Vision Assistant. A CSV file of these extracted features is prepared along with the manual labelling of crack and non-crack. The crack features are as 1 while the non-crack features are marked as 0. The entire dataset consists of 255 crack data and 230 non-crack data. The train-test split is considered as 0.75 and 0.25 respectively. Training, testing and performance evaluation of the SVM classifier are carried out similarly using the prepared dataset. For the SVM classifier model, the value of C is 10 showed the best results amongst the other values of C and Radial Basis Function (RBF) kernel is used. The value of C is identified after multiple iterations of training on the dataset. As compared to the other kernel functions, RBF kernel provides better results and hence is used for the implementation. The data is normalized using the min-max scaler because using the unnormalized features, the SVM classifier showed an accuracy of 0.98 in training while an accuracy of 0.60 is achieved in testing. The model is showing overfitting and to overcome the issue, data normalization was used. After normalization, the classier shows a training accuracy of 0.89 and a testing accuracy of 0.84. Figure 58 shows the training and testing accuracy as viewed in the screenshot.



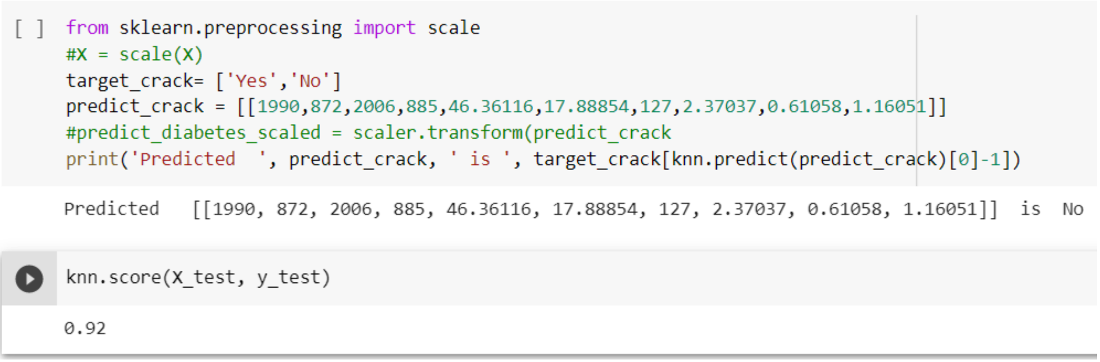
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

clf = SVC(C=10).fit(X_train_scaled, y_train)
print('Crack dataset (normalized with MinMax scaling)')
print('RBF-kernel SVC (with MinMax scaling) training set accuracy: {:.2f}'
     .format(clf.score(X_train_scaled, y_train)))
print('RBF-kernel SVC (with MinMax scaling) test set accuracy: {:.2f}'
     .format(clf.score(X_test_scaled, y_test)))
```

```
Crack dataset (normalized with MinMax scaling)
RBF-kernel SVC (with MinMax scaling) training set accuracy: 0.89
RBF-kernel SVC (with MinMax scaling) test set accuracy: 0.84
```

Figure 58. Result of classification using SVM.

Cross-validation is used in the KNN algorithm and for k=3, the highest training accuracy of 0.96 is achieved. Using the k=3, testing is carried out on the data wherein the test accuracy of 0.93 is reported. The trained model is used to evaluate the classification results on unknown feature data extracted from additional images of concrete cubes. An accuracy of 0.92 is reported in the evaluation of the KNN model. Figure 59 shows the screenshot of the python notebook code from Google Colab. The model evaluation can be seen along with the testing score.



```
from sklearn.preprocessing import scale
#X = scale(X)
target_crack= ['Yes','No']
predict_crack = [[1990,872,2006,885,46.36116,17.88854,127,2.37037,0.61058,1.16051]]
#predict_diabetes_scaled = scaler.transform(predict_crack
print('Predicted  ', predict_crack, ' is ', target_crack[knn.predict(predict_crack)[0]-1])
```

```
Predicted   [[1990, 872, 2006, 885, 46.36116, 17.88854, 127, 2.37037, 0.61058, 1.16051]]  is  No
```

```
knn.score(X_test, y_test)
```

```
0.92
```

Figure 59. Result of classification using KNN.

Out of the two methods tried and tested for crack detection, KNN performs better which is evident from the results. Both the training and testing accuracy of the KNN method is much better than the accuracies of SVM and hence further detailed evaluation of the KNN method with k=3 is performed. Figure 60 shows the graph of the accuracy score versus the training set size for the KNN method. The curve of the training score is shown with a dashed line while the cross-validation score is shown with a solid line. Both the scores are above 0.90 which encourages the performance evaluation.

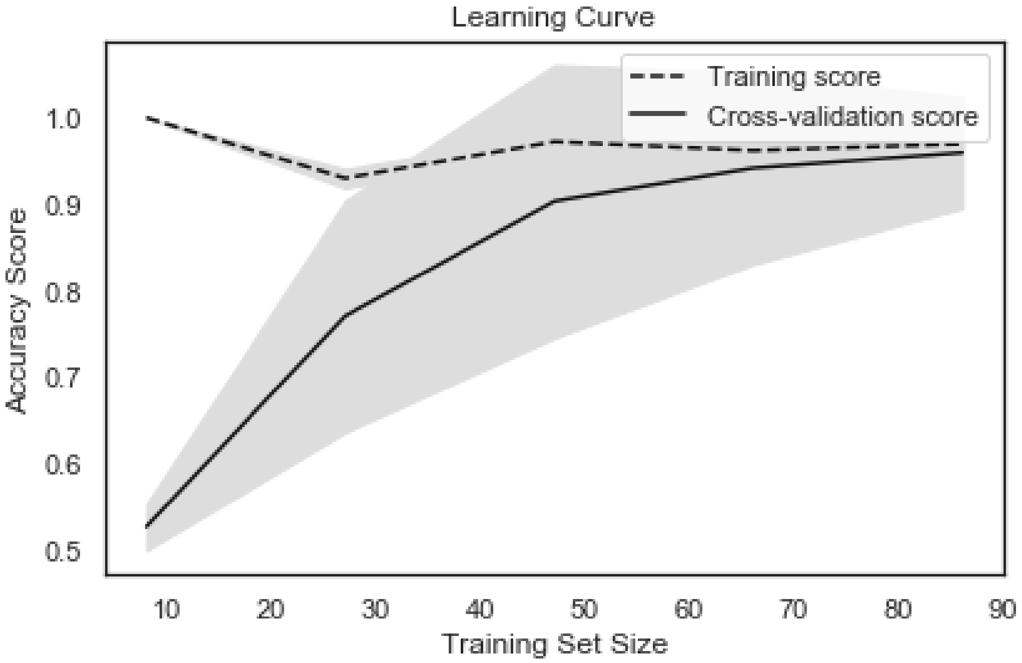


Figure 60. Learning curve of KNN.

Looking into the performance of the KNN method over training and testing data, further performance evaluation is carried out which is presented in Table 4. The analysis comprises recall, precision, accuracy and F score. In the binary classification problem considered here, cracks are considered a positive class while non-cracks are considered a negative class. C shows the number of crack contours obtained in the segmented image while NC shows the number of non-crack contours. The number of crack regions classified correctly as cracks by the model is shown by True Positives (TP), the number of cracks classified incorrectly as non-cracks is shown by False Negatives (FN), the number of non-cracks correctly classified as non-cracks is shown by True Negatives (TN) and the number of non-crack regions incorrectly classified as cracks are shown by False Positives (FP) in Table 4. Recall (R) shows the sensitivity of the model which is calculated by taking the ratio of true positive to the sum of true positive and false negative. It is a factor which indicates the number of times the model incorrectly classified a crack region as a non-crack. Precision (P) indicates the specificity of the model which is calculated by calculating the ratio of true positive to the sum of true positive and false positive. The factors show the number of times the model incorrectly classified a non-crack region as a crack.

Table 4. Performance analysis of classifier using KNN.

Image	C	NC	TP	FN	TN	FP	R	P	A	F
1	23	204	7	16	200	4	0.30	0.64	0.91	0.41
2	14	159	8	6	158	1	0.57	0.89	0.96	0.70
3	25	158	12	13	158	0	0.48	1.00	0.93	0.65
4	32	139	24	8	135	4	0.75	0.86	0.93	0.80
5	11	109	11	0	96	13	1.00	0.46	0.89	0.63
6	18	52	16	2	39	13	0.89	0.55	0.79	0.68
7	12	65	9	3	54	11	0.75	0.45	0.82	0.56
8	33	140	11	22	130	10	0.33	0.52	0.82	0.41
Average	**21.00**	**128.25**	**12.25**	**8.75**	**121.25**	**7.00**	**0.63**	**0.67**	**0.88**	**0.60**

Accuracy (A) indicated the overall efficiency of the trained model and is computed by taking the ratio of the sum of true positive and true negative to the sum of positives and negatives. F score is also called as the harmonic mean of recall and precision which is calculated by taking the ratio of two times the multiplication of precision and recall to the sum of precision and recall. Ideally, the model should achieve a 1.0 value as recall, precision, accuracy and F score. In addition to that, the number of false positives and false negatives should be zero. However, referring the quantitative evaluation data obtained in the performance analysis shows the model classification results are not as desired.

Multiple compression tests are conducted on different concrete cubes. Out of these multiple tests data, few of the image data were utilized for model evaluation post satisfactory training, validation and testing. The eight-image data shown in the Table 4 comprises of the images taken from different iterations of the test. Each image was taken from different concrete cube and possess different characteristics in terms of grey level intensities, random dents, noises present on the surface, crack occurrence and progress with load. The collected data is prepared in a similar way as that of the training and testing data. These images are processed and features of all crack and non-cracks regions are extracted using the particle analysis tool.

Based on the value of the threshold chosen, different numbers of regions are segmented and their feature data are extracted. The number of regions which are segmented in the concrete cube image varies from a minimum value of 70, the maximum value of 227 and the average value of 149.25. The average number of crack regions obtained in these images is reported as 21 out of which around 12 crack regions are correctly classified as cracks by the trained model. On the other hand, the average number of non-crack regions is reported to be around 128 out of which 121 non-crack regions are correctly classified as non-cracks. The average values of recall, precision, accuracy and F score are reported as 0.63, 0.67, 0.88 and 0.60 respectively.

The primary interpretation which can be drawn from the overall performance is that the model is showing signs of underfitting as the training accuracy of 0.96 is achieved. The practical accuracy of the binary classification model to predict a contour as crack or non-crack can be considered 0.99. In addition to that, the testing accuracy of the model is 0.93 and the average accuracy in the performance evaluation is 0.88 which is a clear indication of the overfitting. One of the probable reasons can be the lack of a higher amount of data for training and testing. The other major cause can be features selected to classify crack and non-crack may not be enough and distinguishable for accurate classification. As the features are extracted using a binary image which only contains shape-based information, the extracted features do not contain sufficient information for better classification. Considering the interpretation of the performance evaluation, the use of the DL methods for the classification of crack and non-crack contours seems a good solution. The inherent feature extraction in the DL methods is an additional advantage along with better classification accuracy.

4.6.4 Performance evaluation of convolutional neural network

The performance of CNN has been impressive as compared to conventional image processing-based algorithms. However, to understand the said improvement quantitatively, performance evaluation and interpretation of the trained networks are very important. The performance metrics defined at the beginning of the discussion on the CNN are utilized for assessment. The present work is evaluated in terms of crack detection accuracy and crack monitoring performance. The machine vision-based integrated system developed in the present work is evaluated for crack detection accuracy, precision, recall and F score. The evaluation is carried out to understand the performance of a CNN for crack detection. The assessment is carried out using a dataset prepared using binary contour images, greyscale contour images with data imbalance and greyscale contour images with data augmentation.

The techniques to overcome overfitting like data augmentation, dropout, batch normalization, and regularization are used in the model. In addition to that, hyperparameters like epoch, batch size, number of steps per epoch, and learning rate were selected after multiple iterations. The results obtained for the training, validation and evaluation show that model is not overfitting. In addition to that, a comparison of crack detection using ML and CNN is presented. In order to justify the performance of the present crack detection methodology, a detailed comparison with the other methods from the literature is carried out.

4.6.4.1 Binary contour images

Utilizing the segmented output of the concrete cube image, a crack and non-crack contour image dataset is prepared. The binary contour image dataset comprises 18,296 numbers of non-crack contours and 3,468 numbers of crack contours. The dataset is used for the satisfactory training of the inception v3 model. Post-training, the model is evaluated using unseen images of crack and non-crack contours. For the evaluation, 15 full images of different concrete cubes are used. Table 5 shows the testing evaluation of the trained inception v3 model with the binary dataset. The table comprises of analysis of 15 full images of concrete cubes. The last row in the table shows the average of each column. The analysis comprises the number of crack contours (C), the number of non-crack contours (NC), true positives (TP), false positives (FP), true negatives (TN), false negatives (FN), recall (R), precision (P), accuracy (A) and F score (F).

Table 5. Performance evaluation of CNN tested on a binary dataset.

Image	**C**	**NC**	**TP**	**FN**	**TN**	**FP**	**R**	**P**	**A**	**F**
1	11	186	8	3	186	0	0.73	1.00	0.98	0.84
2	4	41	3	1	39	2	0.75	0.60	0.93	0.67
3	13	156	10	3	154	2	0.77	0.83	0.97	0.80
4	15	130	12	3	126	4	0.80	0.75	0.95	0.77
5	9	165	8	1	161	4	0.89	0.67	0.97	0.76
6	11	135	8	3	131	4	0.73	0.67	0.95	0.70
7	14	174	11	3	171	3	0.79	0.79	0.97	0.79
8	9	38	9	0	37	1	1.00	0.90	0.98	0.95
9	14	162	12	2	158	4	0.86	0.75	0.97	0.80
10	15	133	11	4	128	5	0.73	0.69	0.94	0.71
11	14	162	12	2	158	4	0.86	0.75	0.97	0.80
12	15	133	11	4	128	5	0.73	0.69	0.94	0.71
13	11	162	9	2	159	3	0.82	0.75	0.97	0.78
14	14	124	9	5	120	4	0.64	0.69	0.93	0.67
15	17	94	13	4	89	5	0.76	0.72	0.92	0.74
Average	**12.40**	**133.00**	**9.73**	**2.67**	**129.67**	**3.33**	**0.79**	**0.75**	**0.96**	**0.77**

As observed in Table 5, the testing images contained 12 crack contours and 133 non-crack contours on average. Out of the 12 crack contours, around 10 contours have correctly classified a crack whereas 2 crack contours are misclassified as non-crack contours. Similar incorrect classification is observed in the case of non-crack contours as well. The expected performance of the inception v3 model when trained on a binary contour image dataset is to achieve a minimum number of false positives and false negatives. However, few crack contours are identified as non-cracks and few non-crack contours are predicted as cracks by the trained inception v3 model. Although the model achieved an average accuracy of 0.96, the average values of precision and recall are 0.75 and 0.79 respectively. The average F score is also 0.77.

The average values of accuracy are higher as compared to F-score because accuracy considers the total true predictions versus the total number of predictions. In current problem, the number of true positives and true negatives is very high as compared to false positives and false negatives. The F-score values are lower as compared to accuracy because the number of true positives is smaller in number and used to calculate the recall and precision which governs the F-score. It is evident that the obtained values of average precision and recall are not matching the expected values. It signifies that the sensitivity and specificity of the trained inception v3 model are not as expected. One of the reasons for such results can be the similarity in the shapes of minor cracks and dents, noises which are treated as non-cracks. Because binary images do not have intensity information, the trained inception v3 model shows signs of overfitting because it is not able to generalize the test data in terms of precision and recall.

Figure 61 shows the result of crack detection in two concrete cube samples. Subfigures (a) and (c) show the two concrete cubes and sub-figures (b) and (d) shows the crack detection overlayed result images. The detected cracks are highlighted with green coloured bounding box whereas the non-crack contours are highlighted by a blue coloured bounding box. Careful observation of both result images depicts that the trained inception v3 model is overfitting. Multiple non-crack contours are classified and shown as crack contours in both images.

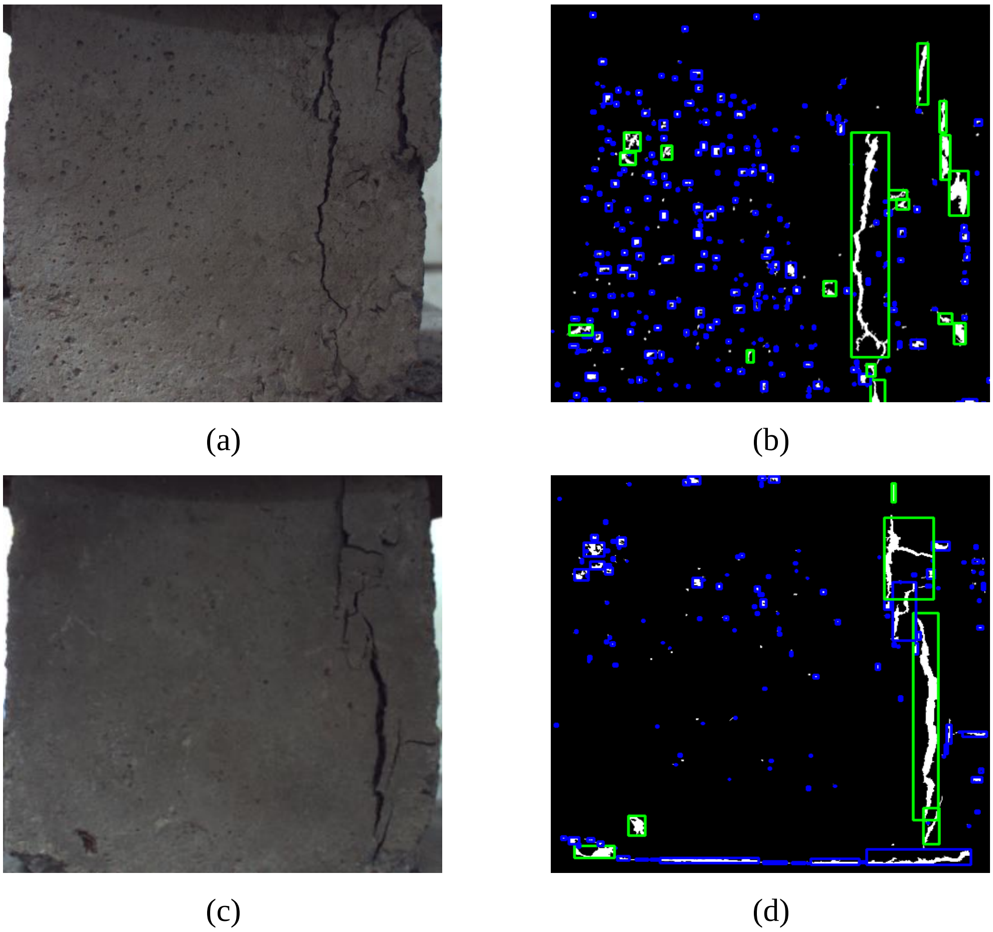

(a)
(b)
(c)
(d)

Figure 61. Result images of crack detection in concrete cubes.

From the presented parametric evaluation and result images, it can be concluded that the trained inception v3 model shows good accuracy in crack detection. However, the average score of recall, precision and F score is not as expected and the model shows signs of overfitting. Some the non-crack contours are identified as cracks by the model. The assessment of the greyscale contour image dataset is presented in the next section.

4.6.4.2 Greyscale contour images

The performance of CNN has been impressive as compared to conventional image processing-based algorithms. However, to understand the said improvement quantitatively, performance evaluation and interpretation of the networks are extremely important. A trained CNN should be tested for performance to judge its accuracy when used in a practical application. The evaluation will justify whether the model is suitable or it requires modification (Hongyu Zhu , 2018). The developed crack detection methodology work is evaluated in terms of crack detection accuracy and crack monitoring performance. The machine vision-based integrated system developed in the present work is evaluated for crack detection accuracy, precision and recall. The evaluation is carried out to understand the performance of a CNN for crack detection. Two instances of the inception v3 model are evaluated. One of the instances is the model trained with an imbalanced greyscale contour image dataset whereas the second instance is the model trained with a balanced greyscale contour image dataset.

Table 6 and As observed in Table 6, the performance of the trained inception v3 model seems satisfactory considering the average accuracy which is 0.98. The testing of performance is carried out using 15 full images of the concrete cubes which are not used for training and validation. The flowchart of model testing as shown in Figure 43 is used for the performance evaluation. The testing images contained an average of 9 crack contours and 170 non-crack contours. Around 7 crack contours are accurately classified as cracks while the remaining 2 crack contours are wrongly classified as non-cracks. Similarly, on average 169.93 non-crack contours are correctly classified as non-cracks and 0.27 non-crack contours are classified as cracks. It is evident from these numerical results that the model has perfectly classified non-crack contours. However, few crack contours are misclassified as non-cracks which shows poor specificity or recall.

Even though the trained inception v3 CNN model has achieved an average accuracy of 0.98, the average values of precision, recall, and F score are 0.96, 0.80, and 0.86 respectively. The average value of recall is significantly low and does not meet the expected performance. One of the substantial reasons is the data imbalance. Non-crack contours are 18,296 in number whereas the crack contours are 3,468 which shows a clear data imbalance. This imbalance created a bias in the learning of the neural network which is evident from the results depicted in the values of recall and precision. The model does not make mistakes in the classification of non-crack contours. On the other hand, the same model does multiple wrong classifications in the case of crack contours.

The trained inception v3 model shows expected performance in the classification of non-crack contours because the precision obtained in the analysis is 0.96. This also indicates that the specificity of the model is high and as expected. Precision is also known as specificity and signifies how many non-crack contours are wrongly classified as crack contours. The expected recall is not achieved due to the data imbalance. The recall is also known as the sensitivity of the model and it shows how many crack contours are incorrectly classified as crack contours. Due to poor recall and high precision obtained during the performance evaluation of the trained inception v3 model, the average value of the F score which is also known as the harmonic mean of precision and recall is 0.86.

Table 7 show the performance of these two instances of the inception v3 CNN model. Both tables encompass the testing data carried out on 15 different concrete cube images along with the average of all 15 images for each parameter. The table presents the number of crack contours (C), number of non-crack contours (NC), number of True Positives (TP), number of False Negatives (FN), number of True Negatives (TN), number of False Positives (FP), Recall (R), Precision (P), and Accuracy (A). The parameters used for the evaluation are a standard in CNN-based object detection and classification applications (Ali Raza Asif , 2020) (Chaitanya Nagpal , 2019) (Zewen Li , 2021).

In the present study, crack is considered a positive class whereas non-crack is considered a negative class. The testing is carried out manually using the concrete cube images which are not part of the training data set. The ideal values of FP and FN for an accurate CNN model should be zero. Accuracy is represented by A and it indicates the overall effectiveness of the classifier. The parameter precision and recall represented by P and R respectively show the goodness of the network to classify the classes in terms of sensitivity and specificity. For an ideal network, the values of precision and recall should be 1 and the value of FP and FN should be 0. The average accuracy of 98% is achieved during the testing with good precision and high recall. An actual crack is predicted as non-crack very few times, which shows the model has a high recall. The number of times, the trained model predicted an actual non-crack as a crack is shown by precision.

Table 6. Performance evaluation of CNN tested on imbalanced greyscale dataset.

Image	**C**	**NC**	**TP**	**FN**	**TN**	**FP**	**R**	**P**	**A**	**F**
1	8	188	8	0	187	1	1.00	0.89	0.99	0.94
2	8	160	7	1	160	0	0.88	1.00	0.99	0.93
3	7	170	6	1	170	0	0.86	1.00	0.99	0.92
4	7	183	6	1	183	0	0.86	1.00	0.99	0.92
5	15	336	13	2	336	0	0.87	1.00	0.99	0.93
6	5	172	5	0	172	0	1.00	1.00	1.00	1.00
7	8	348	8	0	348	0	1.00	1.00	1.00	1.00
8	5	97	4	1	96	1	0.80	0.80	0.98	0.80
9	7	100	4	3	100	0	0.57	1.00	0.97	0.73
10	10	98	7	3	98	0	0.70	1.00	0.97	0.82
11	6	102	4	2	101	1	0.67	0.80	0.97	0.73
12	10	90	5	5	90	0	0.50	1.00	0.95	0.67
13	12	86	9	3	86	0	0.75	1.00	0.97	0.86
14	18	109	16	2	109	0	0.89	1.00	0.98	0.94
15	15	314	9	6	313	1	0.60	0.90	0.98	0.72
Average	**9.40**	**170.20**	**7.40**	**2.00**	**169.93**	**0.27**	**0.80**	**0.96**	**0.98**	**0.86**

As observed in Table 6, the performance of the trained inception v3 model seems satisfactory considering the average accuracy which is 0.98. The testing of performance is carried out using 15 full images of the concrete cubes which are not used for training and validation. The flowchart of model testing as shown in Figure 43 is used for the performance evaluation. The testing images contained an average of 9 crack contours and 170 non-crack contours. Around 7 crack contours are accurately classified as cracks while the remaining 2 crack contours are wrongly classified as non-cracks. Similarly, on average 169.93 non-crack contours are correctly classified as non-cracks and 0.27 non-crack contours are classified as cracks. It is evident from these numerical results that the model has perfectly classified non-crack contours. However, few crack contours are misclassified as non-cracks which shows poor specificity or recall.

Even though the trained inception v3 CNN model has achieved an average accuracy of 0.98, the average values of precision, recall, and F score are 0.96, 0.80, and 0.86 respectively. The average value of recall is significantly low and does not meet the expected performance. One of the substantial reasons is the data imbalance. Non-crack contours are 18,296 in number whereas the crack contours are 3,468 which shows a clear data imbalance. This imbalance created a bias in the learning of the neural network which is evident from the results depicted in the values of recall and precision. The model does not make mistakes in the classification of non-crack contours. On the other hand, the same model does multiple wrong classifications in the case of crack contours.

The trained inception v3 model shows expected performance in the classification of non-crack contours because the precision obtained in the analysis is 0.96. This also indicates that the specificity of the model is high and as expected. Precision is also known as specificity and signifies how many non-crack contours are wrongly classified as crack contours. The expected recall is not achieved due to the data imbalance. The recall is also known as the sensitivity of the model and it shows how many crack contours are incorrectly classified as crack contours. Due to poor recall and high precision obtained during the performance evaluation of the trained inception v3 model, the average value of the F score which is also known as the harmonic mean of precision and recall is 0.86.

Table 7. Performance evaluation of CNN tested on a balanced greyscale dataset.

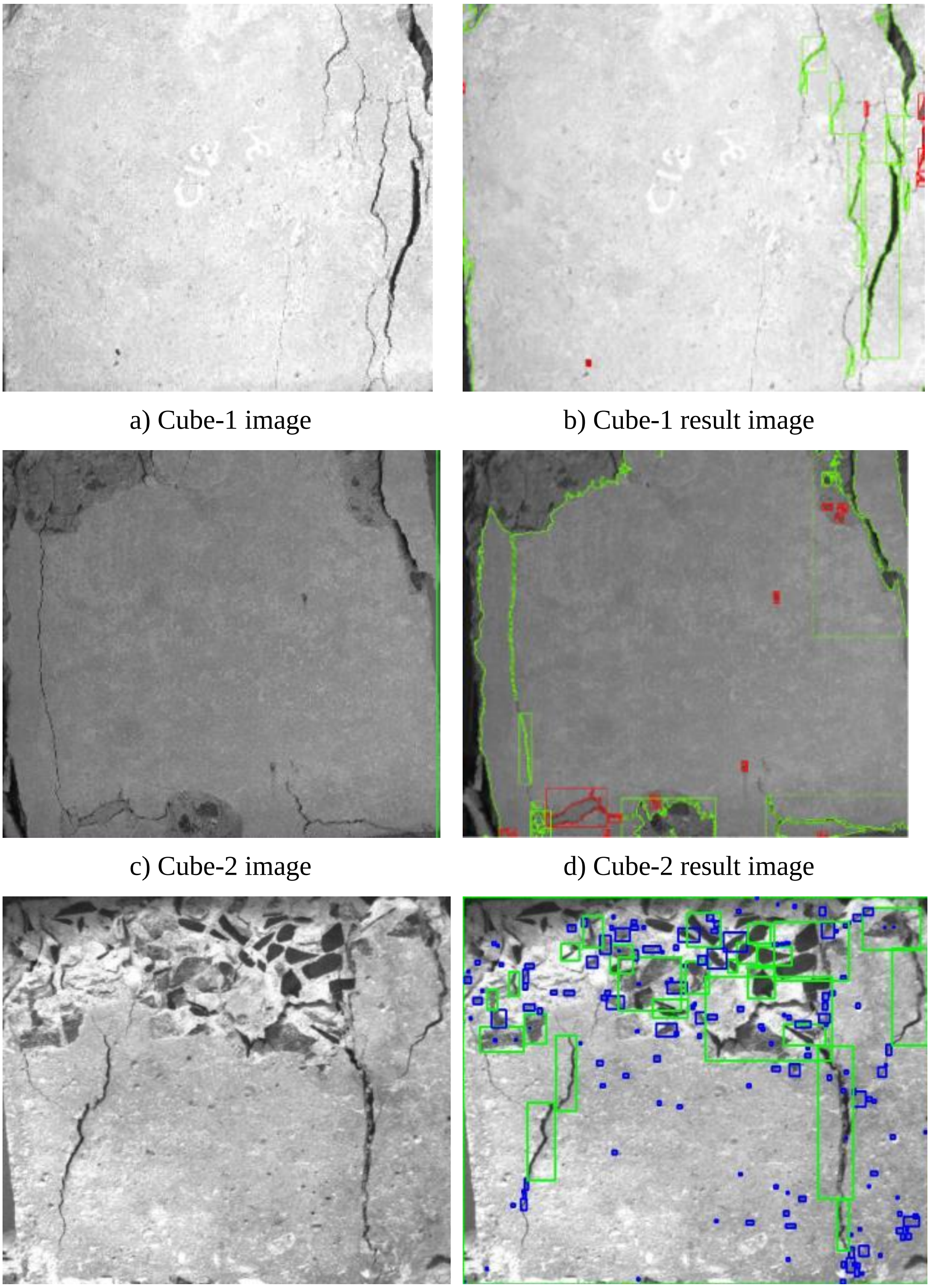
Image	**C**	**NC**	**TP**	**FN**	**TN**	**FP**	**R**	**P**	**A**	**F**
1	7	197	5	2	194	3	0.71	0.63	0.98	0.67
2	13	355	11	2	352	3	1.00	0.81	0.99	0.90
3	10	258	10	0	256	2	1.00	0.83	0.99	0.91
4	10	250	9	1	248	2	1.00	0.83	0.99	0.91
5	10	255	9	1	253	2	1.00	0.83	0.99	0.91
6	4	193	4	0	193	0	1.00	1.00	1.00	1.00
7	7	227	6	1	227	0	1.00	1.00	1.00	1.00
8	6	191	5	1	191	0	1.00	1.00	0.99	1.00
9	7	232	6	1	232	0	1.00	1.00	1.00	1.00
10	6	267	5	1	267	0	1.00	1.00	1.00	1.00
11	6	253	4	2	252	1	1.00	0.86	0.99	0.92
12	9	169	8	1	166	3	1.00	0.75	0.98	0.86
13	11	146	7	4	146	0	1.00	1.00	0.97	1.00
14	9	164	6	3	163	1	1.00	0.90	0.98	0.95
15	9	171	7	2	170	1	1.00	0.90	0.98	0.95
Average	**8.27**	**221.87**	**6.80**	**1.47**	**220.67**	**1.20**	**0.98**	**0.88**	**0.99**	**0.93**

A significant difference is observed between the average values of accuracy and F-score. The difference is due to the values of recall which are not good as compared to overall accuracy and precision. As seen in the table, the values of false negatives are higher and significant as compared to the number of true positives. Due to this fact, the values of recall are poor and F-score as well. The results obtained during the evaluation signify the effect of data imbalance as well as the overall performance improvement as compared with the results obtained using the binary contour image dataset.

Table 7 shows the performance evaluation of the inception v3 CNN model trained using a balanced greyscale contour image dataset. The problem of data imbalance is addressed using data augmentation. After the augmentation process, around 80,000 images per class are used for the training of the inception v3 model.

The performance analysis is carried out on 15 full images of different concrete cubes acquired using the experimental setup. These images are chosen to consider different instances during the compression tests of concrete cubes. Due to this, the number of crack and non-crack contours present in each of these images is different. The minimum and the maximum number of crack contours found in these test images are 4 and 13 respectively. On the other hand, the minimum and the maximum number of non-crack contours found are 146 and 355 respectively. Out of the 8.27 average number of crack contours, 6.80 crack contours are correctly classified as crack whereas 1.27 crack contours are misclassified as non-cracks. On the other hand, 220.67 non-crack contours are correctly classified as non-cracks and the remaining 1.20 non-crack contours are erroneously classified as cracks. The average values of accuracy, recall, precision and F-score are 0.98,0.88,0.99 and 0.93 respectively. The obtained performance meets the expectation. In addition to that, the trained model also overcomes the issues observed in the analysis of the imbalanced datasets and binary datasets.

Crack detection results can be observed by comparing the input image and result image with detected cracks. Figure 62 shows the sample image of concrete cubes and crack detection result images. The results are overlayed on the images with a green colour bounding box for cracks and a blue and/or red colour bounding box for non-crack contours. The images of the cube are acquired during the compression testing on concrete cubes. The developed methodology performed accurate crack detection which is evident from the result images.



a) Cube-1 image

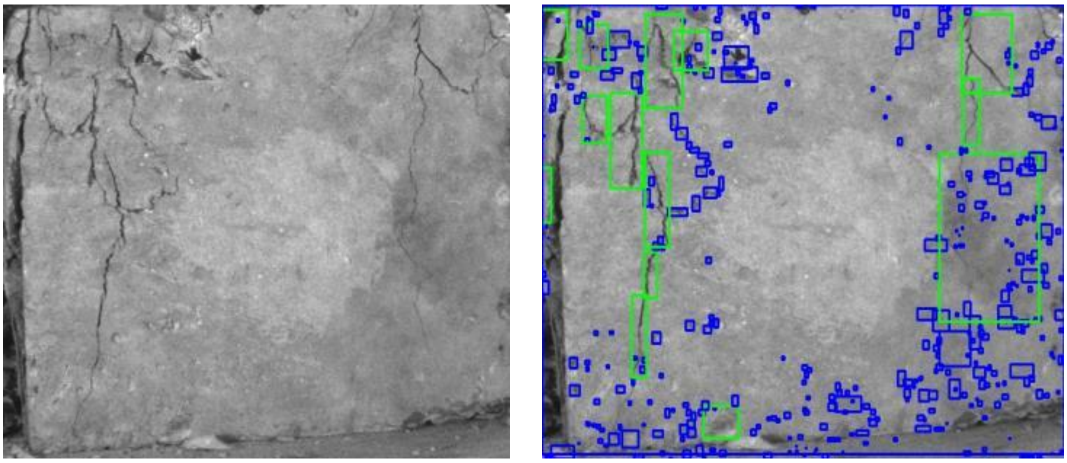
b) Cube-1 result image

c) Cube-2 image

d) Cube-2 result image

e) Cube-3 image

f) Cube-3 result image



g) Cube-4 image h) Cube-4 result image

Figure 62. Sample images of concrete cubes and their results.

From the results obtained during analysis and crack detection images, it can be concluded that the data augmentation process significantly improves the results of precision, recall and F-score. In addition to that, the inception v3 CNN model trained using the balanced and augmented greyscale contour image dataset overcomes all the issues and challenges reported in the case of the binary and imbalanced greyscale datasets. Comparative analysis with all the implemented methods and methods presented in the literature is discussed in the next section of the chapter.

4.6.4.3 Comparative assessment of implemented methods

The comparative analysis of different methods used for crack detection in the concrete cube is presented to understand the performance of each method with respect to the other. All the three CNNs are trained using inception v3 model with similar train-validation-test split. However, batch size, number of batches and epochs are varied. The best values of hyperparameters are listed in the Table 8. In addition to that, the details of dataset and number of images in the dataset are also mentioned.

Table 8 Hyperparameters of CNNs.

Method	Dataset type and size	Model, train-validation-test split, batch size, number of batches, epoch, learning rate
CNN1	Binary images, class imbalance, 3468 crack images and 18296 non-crack images	Inceptionv3, 80-10-10, 100, 218, 5000, 0.01
CNN2	Grayscale images, class imbalance, 3468 crack images and 18296 non-crack images	Inceptionv3, 80-10-10, 100, 218, 5000, 0.01
CNN3	Grayscale images, class balance, 80000 crack and 80000 non crack	Inceptionv3, 80-10-10, 100, 1600, 5000, 0.01 (adaptive learning rate)

Table 9 showcase the comparison between different methods which are implemented in the present work. The comparison is analysed based on crack detection performance. The average values of each parameter are depicted in the comparison table. For the comparison, the ML method, and three instances of the inception v3 model are used.

Table 9. Comparative analysis of crack detection performance of implemented methods.

Method	C	NC	TP	FN	TN	FP	R	P	A	F
ML	21.00	128.25	12.25	8.75	121.25	7.00	0.63	0.67	0.88	0.60
CNN1	12.40	133.00	9.73	2.67	129.67	3.33	0.79	0.75	0.96	0.77
CNN2	9.40	170.20	7.40	2.00	169.93	0.27	0.80	0.96	0.98	0.86
CNN3	8.27	221.87	6.80	1.47	220.67	1.20	0.98	0.88	0.99	0.93

The three instances are CNN1 which is showing the performance of the inception v3 model when trained on a binary contour image dataset, and CNN2 is the performance of the same model when trained on an imbalanced greyscale contour image dataset.

The performance of the best model is shown by CNN3 which is the model trained on a balanced and augmented greyscale contour image dataset. The number of cracks and non-crack contours varies significantly which is due to the randomness of dents, and noises present on the surface of the concrete cube. The randomness of cracks occurring in different concrete cubes due to the compressive load also affects the number of crack contours segmented in the image. From the comparison table, it is evident that the CNN3 inception v3 model shows the best performance in terms of accuracy, precision, recall and F score. The performance of the model is good enough to be deployed in real-time crack detection in concrete cubes during the compression tests. In order to prove the performance of the present crack detection methodology when compared to the crack detection approaches presented in the literature, a comparison analysis is discussed in the next section.

4.6.4.4 Comparison with methods presented in the literature

The results obtained with the current methodology are compared and analysed with other methodologies from the literature. The comparison is performed with the literature methods wherein CNN are used. Table 10 shows the comparison of different methods proposed by authors based on the model's testing accuracy along with other factors such as crack surface, CNN model used, image acquisition setup and dataset used in the work. The majority of the work has been carried out on pavement images of different locations, buildings wall images and concrete bridge images.

Based on the CNN model average accuracy, it is evident from the comparison that the proposed method obtained the best results. Other models like MatConvNet, ConvNet, modified AlexNet, VGG-16 and a few other customized CNNs obtained good results (Shengyuan Li , 2019) (N. A. M. Yusof , 2019) (Zhang Lei , 2016) (Gopalakrishnan , 2017) (Pauly , 2017) (Hongyan Xu , 2019). Some of the authors have used smartphones and DLSRs for image acquisition of cracks at different locations. For pavements, authors have focused on the use of off-the-shelf 3D vision systems. None of the authors has included the synchronization of cracks with the applied load which is performed in the present work. Since 2015, the use of CNN and other related methods has been observed in the literature.

Table 10. Performance comparison of different methods from the literature.

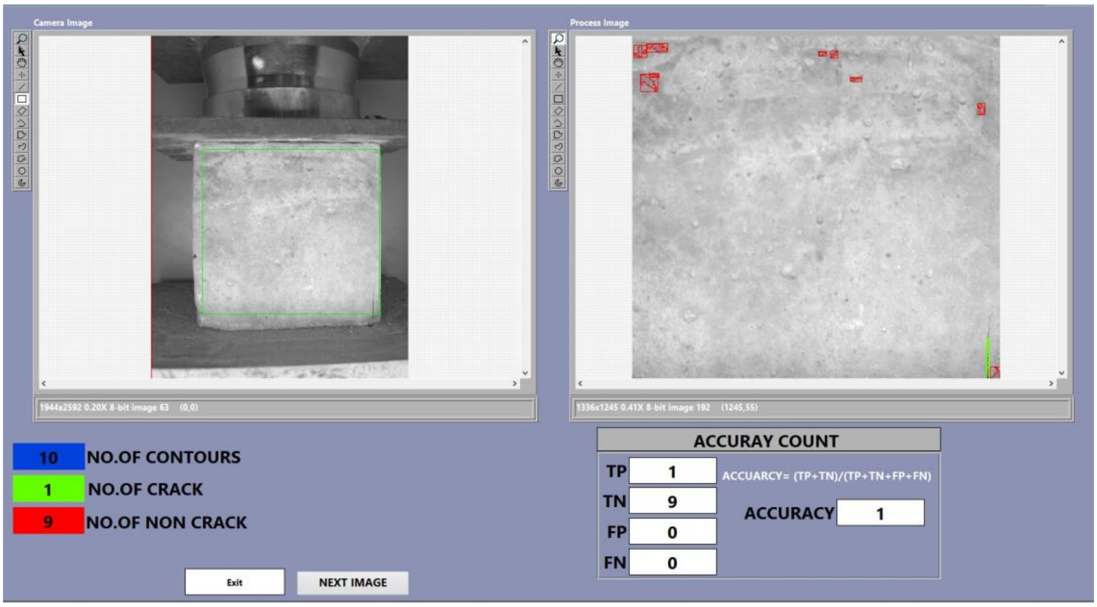
Author and year	**Surface for crack detection**	**CNN model**	**Results**	**Image acquisition apparatus and dataset**	**Synchronization of crack with loading**
Zhang Lei et. al. (Zhang Lei , 2016)	Pavements at Temple University, USA	ConvNet	P = 0.87, R = 0.93, F = 0.90	Smartphone, Custom dataset	Not included
Young-Jin Cha et. al. (Young-Jin Cha , 2017)	Complex engineering building at University of Manitoba, Canada	MatConvNet	A = 0.98	Hand-held Nikon D5200 DSLR, Custom dataset	Not included
Gopalakrishnan et al. (Gopalakrishnan , 2017)	Hot-Mix Asphalt (HMA) and Portland Cement Concrete (PCC) surfaced pavement	VGG-16	A = 0.90, P = 0.90, R = 0.90, F = 0.90	Federal High Way Administration (FHWA) /Long Term Pavement Performance (LTPP) database	Not included
Pauly et. al. (Pauly , 2017)	Pavement images of premises of Temple University, USA	Custom CNN	A = 0.90, P = 0.92, R = 0.88	Smart phones, Custom dataset	Not included
Allen Zhang et. al. (Allen Zhang , 2018)	3D pavement images	CrackNet II	P = 0.90, R = 0.89, F = 0.90	PaveVision3D Ultra, Custom dataset	Not included
Hongyan Xu et. al. (2019) (Hongyan Xu , 2019)	Concrete Bridge	Custom CNN	A = 0.96, P = 0.78, F = 0.88	Phantom 4 Pro's CMOS surface array camera, Custom dataset	Not included

Author and year	**Surface for crack detection**	**CNN model**	**Results**	**Image acquisition apparatus and dataset**	**Synchronization of crack with loading**
Shengyuan Li et. al. (Shengyuan Li , 2019)	Bridge towers and anchor chambers of a suspension bridge in Dalian, Liaoning, China	Modified AlexNet	A = 0.99	Smartphone, Custom dataset	Not included
N A M Yusof et. al. (N. A. M. Yusof , 2019)	Asphalt pavement	Custom CNN	A = 0.99, P = 0.99, R = 0.98	NIKON digital camera, Custom dataset	Not included
Proposed Method	Laboratory-scale concrete cube	Inception v3	A = 0.99, P = 0.88, R = 0.98, F = 0.93	Basler industrial camera, Custom dataset	Included

From the numerical analysis reported in Table 7 and the performance comparison of different methods mentioned in Table 10, the proposed method shows overall improved performance for concrete crack detection in laboratory concrete cubes.

4.6.5 Software implementation

The graphical user interface developed in the present work shows acquired images, and processed images with results overlayed as green and red bounding boxes for crack and non-crack contours respectively, numerical results, and other required information. The interface developed in LabVIEW software is shown in Figure 63. The images are processed, transferred to the CNN, and crack, non-crack classification results are displayed continuously during the compression testing of the concrete cube. The developed system is tested for standard cubes of different compressive strengths. The cubes are prepared as per Indian standards and are allowed to cure for 28 days. All the tests are performed using the compression testing setup.


Camera Image
Process Image
1944x2592 0.20X 8-bit image 63 (0,0)
1336x1245 0.41X 8-bit image 192 (1245,55)
Exit
NEXT IMAGE

10 NO.OF CONTOURS

1 NO.OF CRACK

9 NO.OF NON CRACK

ACCURAY COUNT

TP	1
TN	9
FP	0
FN	0

ACCUARCY= (TP+TN)/(TP+TN+FP+FN)

ACCURACY 1

Figure 63. Concrete crack detection software interface.

During experiments, the number of cracks, length of the cracks, and the area of cracks for a cube specimen are recorded with respect to the applied load. The length and area of a crack are obtained in terms of pixels which is the unit in an image. All the cracks occurring on the monitored surface of the concrete cube are detected and their properties are extracted for analysis. Developed crack detection methodology considers a crack as a contour in the image. The detected crack contour is used to extract the crack length which is the contour length and the crack area which is the contour area. However, to derive an observation and physical relevance, the crack length and crack area extracted in terms of pixels are calibrated and converted into mm and mm^2.

Load versus the number of cracks, load versus length of cracks, and load versus area of cracks graph are plotted to demonstrate the capabilities of the proposed system. Further load at the first crack, the maximum number of cracks, and the corresponding load for all concrete cubes are compared to understand the effectiveness of the present system. The failure pattern of all concrete specimens is monitored during testing to understand the crack propagation using the system developed in this work. Loading of specimens is stopped when the specimen is unable to take further load due to excessive cracking. Real-time graphs of load versus time, load versus the number of cracks, load versus the length of cracks and load versus the area of cracks are also prepared by the software which is shown in Figure 64.

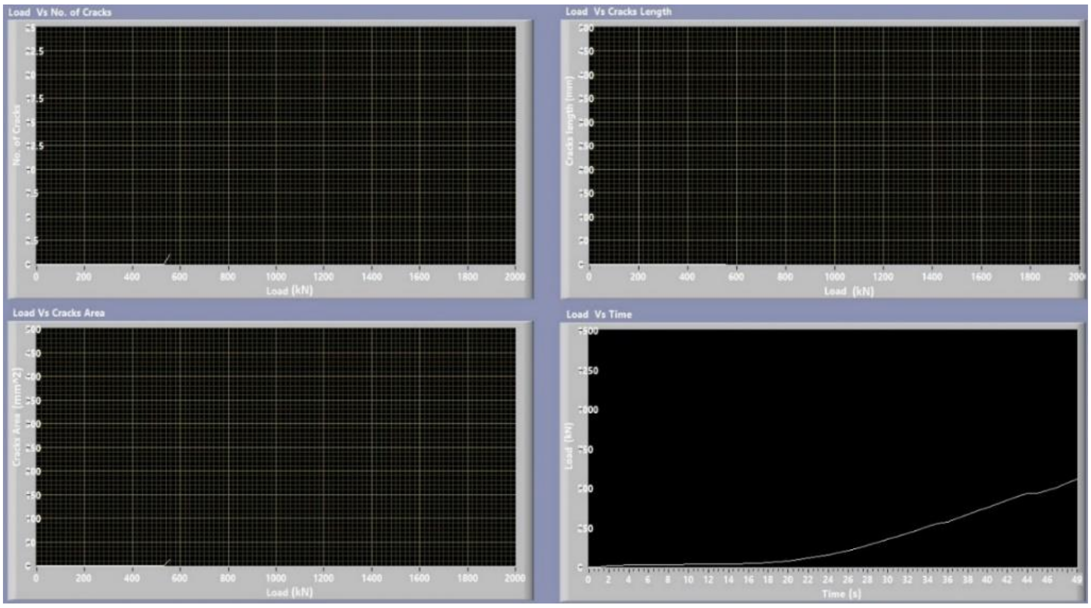


Figure 64. Crack monitoring graphs with respect to load.

The software performs image acquisition, load cell data acquisition, image pre-processing, contour image classification as crack or non-crack, and overlays the result. LabVIEW is used along with Python to perform the above-mentioned tasks. The vision hardware and developed software are seamlessly integrated to perform crack monitoring and analysis during the compression testing of a concrete cube. Acquired greyscale images and processed images with results overlayed in green and red bounding boxes can be seen in the interface screenshot. Computation times, load value, image number, the total number of contours and predicted results are shown in numerical form.

4.6.6 Results of real-time implementation

In order to achieve real-time crack detection and crack-load monitoring, the implementation of real-time crack detection is vital. The implementation of a crack detection classifier is carried out using LabVIEW and Python-based software developed in the present work. The overall objective of the work is focussed on the accuracy of the crack detection. However, the time factor is also considered as the developed experimental system is used online during the compression tests. Here, the data (image and load) sampling rate of 1 sample per second is implemented and the same is sufficient for the detailed failure analysis of concrete cube subjected to compressive loading. This data rate is achieved due to proper selection of hardware components, software and system integration. The number of image and load data captured during a compression test cycle varies from 120 to 270 with a sampling time of 1 data per second. The loading rate of the compression test is kept at 5.25 kN/s for all the test conducted, reported and cited in the text. The variation in the number of images captured for each concrete cube is related to the strength of the concrete cube or the amount of load resisted by the concrete cube. On an average, a total of 8 cracks occurs on the concrete cube surface. The rate of data capturing seems good enough to monitor the progress of cracks occurring on the concrete cube surface during the test. Data rate faster than this can be achieved using dedicated computing system.

Figure 45 and Figure 46 depicted the operations performed in hardware and software which are marked with relevant symbols. Looking into the computation demand of the crack detection classifier and the execution speed obtained on the CPU alone, a USB accelerator from Google Coral is used in the work. In order to understand the improvements in computation time and evaluate the performance quantitatively, an analysis is performed. For one case only CPU is used for the inference time analysis. Thereafter the analysis is carried out using CPU along with TPU. Table 11 shows the interference time in seconds when the contour classification is executed on the CPU in standalone mode. The CPU was considered in idle state with only necessary programs running and other processes were stopped while measuring the time.

Table 11. Inference time analysis on CPU alone.

Image	**Number of Contours**	**Minimum time (s)**	**Maximum time (s)**	**Average time (s)**	**Total time (s)**
1	10	5.201	17.070	6.740	68.092
2	14	4.945	5.480	5.239	74.232
3	3	5.223	5.464	5.354	16.264
4	7	5.048	5.537	5.295	37.368
5	9	5.060	5.560	5.318	48.200

For a better understanding of the variations in the execution time, the number of contours found in the image, and minimum, maximum and average inference time is analysed. The total time is the sum of the execution time for all the contours in the image. It is observed that the interference times are in seconds and the total time taken for the classification of all contours in an image is more than a minute in a few cases. These inference times are too high in comparison to the 1 image per second execution speed required.

Table 12. Inference time analysis on CPU + TPU.

Image	**Number of Contours**	**Minimum time (s)**	**Maximum time (s)**	**Average time (s)**	**Total time (s)**
1	11	0.002	0.039	0.026	0.602
2	3	0.018	0.029	0.025	0.288
3	10	0.026	0.040	0.031	0.639
4	14	0.017	0.043	0.024	0.759
5	14	0.015	0.036	0.024	0.805

The computation time of classifying contours using CPU along with TPU as a computation resource is shown in Table 12. It is observed that the inference times of each contour are in milliseconds in contrast to the inference time obtained using CPU standalone.

The minimum inference time ranges between 2 to 30 milliseconds whereas the maximum time ranges between 29 to 43 milliseconds. The total time taken by the computation resource for the execution of one image is below 1,000 milliseconds which is evident from the total time results depicted in Table 12. Hence, the use of CPU along with TPU is justified as there is a huge increment in the inference speed. The proposed concept can be further extended to other unobserved surfaces of the concrete cube as the current setup only covers one face of the cube for crack detection. Further, the methodology can be adapted for other important laboratory-scale concrete objects and other potential structure monitoring cases. One major drawback of the use of CNNs is the computation time.

4.6.7 Summary

The deep learning model trained in the current work is able to accurately differentiate between a crack and non-crack present on a concrete cube surface. Here, dents, blemishes, and honeycombs which have similarity in intensity levels as compared to cracks are considered as non-crack parts. Such parts are inherently present in random proportions on the surface of a concrete cube. So, the model is trained to classify a segmented region as a crack or non-crack in current work and the average accuracy of 99% is achieved. Road has different texture, intensity and features as compared to the concrete cube. The model used in the current approach is not tested to distinguish between a real crack and an impression on the concrete road that looks like a crack. Hence, it is likely that the model may not be able to differentiate between a real crack and an impression on the concrete road which looks like a crack. However, such experiments and testing can be performed in the future. For crack detection on roads, researcher have reported use of 3D camera system. Such 3D vision system uses laser projection to determine the height or depth of the object in the area of observation. This system can differentiate between a real crack and an impression on the road which looks like a crack. Multiple contributions are found in the domain by a group of researchers comprising of Allen Zhang, Kelvin C P Wang, Yue Fei, and others. These contributions are discussed at length in the Chapter 2 Literature review, under section 2.4 Artificial intelligence based approaches.

Normally, CNNs require a large number of computing resources containing high-speed processors, high-configuration GPU and lots of memory. Time consumed in the manual preparation of the dataset, data augmentation, and training is considerably high as compared to the conventional image processing methods. Researchers have utilized the computing power of high-configuration processors, GPUs and memory for training, testing and validation of CNN applications. Colab by Google is an open, free and computationally resourceful coding environment which has become popular. Nonetheless, the availability of low-cost off-the-shelf computing resources like Google coral, NVIDIA Jetson nano, USB computing sticks from Intel, and Google USB Accelerator would allow the researchers to develop and deploy these CNN-based applications.

Chapter 5 Crack-load monitoring and analysis

Chapter 5 covers the details of crack-load monitoring data and analysis information. Firstly, the algorithm to measure the crack length and crack area from the crack detected using the CNN is presented. The results include the crack-load monitoring and analysis results presented in the form of an overall crack-load monitoring summary, annotated images, graphs and tabular data with remarks. The analysis is carried out for multiple concrete cube samples and presented along with necessary observations and discussions.

5.1 Overview

Crack monitoring with reference to the applied compressive load for concrete cubes of different strengths is carried out in the present work. The classified results of crack and non-crack contours are utilized for crack monitoring. The contours classified as cracks by the CNN are further processed for crack monitoring with respect to the applied load.

The crack-load analysis is carried out from the real-time information extracted using the current approach. The information comprises parameters like occurrence of cracks, load, length of cracks, area of cracks extracted continuously from the acquired image and load data while the cube is subjected to compressive load. In order to carry out a detailed crack-load analysis and to obtain real-time performance, 1 image per second or faster computation is acceptable performance as far as the current experimental situation and system is concerned. The present system is able to process 1 image data per second. The experiments are performed on a personal laptop with limited configuration for implementation of a convolutional neural network computation. However, in order to improve the computation time, a Tensor Processing Unit (TPU) from Google Coral is used alongside the laptop. The results are mentioned in the sub-topic 4.6.6 Results of real-time implementation. In order improve further, a dedicated and high configuration system with GPU on board can be used for the experimentation in future.

Real-time images and applied load are acquired in a synchronous manner, which governs the true mapping of crack monitoring results with the applied load. Approximately 120 to 270 images of concrete cubes and applied load with a sampling time of 1 second are acquired during the testing. The total time of the test and the number of data samples acquired depend on the rate of loading and strength of the concrete cube. For crack monitoring, the measurement of crack length and the crack area is calculated after processing the features of contours which are predicted as crack contours by the CNN model (Chunyan Shao , 2021) (T. Barkavi , 2019). The contour parameters and their feature extraction like moments, area, perimeter, approximation, and bounding box are widely known and used methods in image processing. The crack contours are the lines or curves in the images that connect all the points that lie on the boundary of a crack and have similar intensity. Crack length can be considered as a contour perimeter while the crack area can be estimated from the contour area.

The contour perimeter gives the total sum of all the pixels lying on the boundary of the contour in terms of pixels, while the contour area gives the total sum of all the pixels lying inside the boundary of the contour in terms of pixels. Both contour perimeter and contour area consider the pixels which are having similar intensity levels. The length and area obtained in terms of pixels are calibrated in real-world units using two-point linear calibration. The crack and load monitoring work presented in the chapter contains the details of the methodology of crack length and area measurement. The calibration of obtained measurement is also discussed along with the image processing method and necessary equations. In addition to that, an analysis of crack properties measured for multiple concrete cubes is presented. Further, crack-load data are analysed for individual crack-related analysis data presented in form of tables, crack-wise highlighted images of concrete cubes and graphs of load versus progression in the crack length of each crack. The graphs of load versus progression in the crack area of each crack are also covered in the analysis.

5.2 Measurement of crack length and crack area

In the present work crack detection is being performed with the use of a CNN. However, to understand and analyse the loading behaviour of concrete cubes subject to a compression test, the measurement of crack properties is of equal importance. The present study is focused on the measurement of crack properties which are visible on the external surface of concrete cubes. As the proposed method uses a single camera for monitoring the cracks on any one of the surfaces of the concrete cube during the compression test, other surfaces are not monitored. Hence, the properties visible on the external surface of the observed face of the concrete cube are extracted. The properties are extracted using conventional image processing methods. The cracks in the surface image of the concrete cube are considered contours. The contours were extracted from the greyscale prepossessed image. The crack contours detected by the model are analysed in the developed software platform to extract crack length and crack area. The greyscale images of the crack were filtered using a 3 × 3 kernel and binarized using a thresholding operation. The resultant binary image is analysed for particles and particle length and area were extracted.

The crack length and crack area extracted using the flow shown in Figure 65 were in pixels. The parameters were calibrated to convert the results into mm and mm^2.

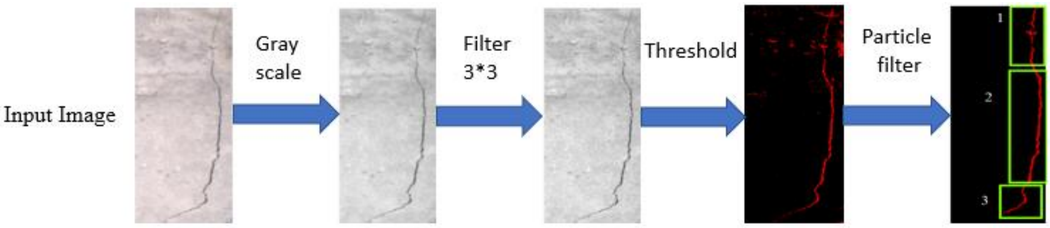


Figure 65. Flow of crack analysis.

For crack length, the pixel coordinates of two corners were extracted to calculate the length of a cube in pixels. The formula in Equation (11) shows the conversion factor which will help in converting pixels into mm.

$$Factor=\frac{Length\ of\ cube\ found\ using\ distance\ formula\ from\ the\ image\ (in\ pixel)}{150\ (cube\ length\ in\ mm)} \quad (11)$$

Particle analysis measures the boundary of the crack, so the resultant measure has to be halved to get the length of the crack. The formula is shown in Equation (12) is used for the conversion.

$$Perimeter(mm) = \frac{Perimeter(in\ pixels)}{2 \times Factor} \ (12)$$

To get the crack area in mm^2 Equations (13), (14) and (15) are used. Initially, the square root of the area in pixel^2 is calculated using formula shown in Equation (13). The result obtained is divided by the factor found in Equation (11). Area of cracks in mm^2 is obtained by taking the square of the result obtained in Equation (14).

$$x = \sqrt{Area\ (in\ pixel^2)} \ (13)$$

$$y = \frac{x}{Factor} \ \ (14)$$

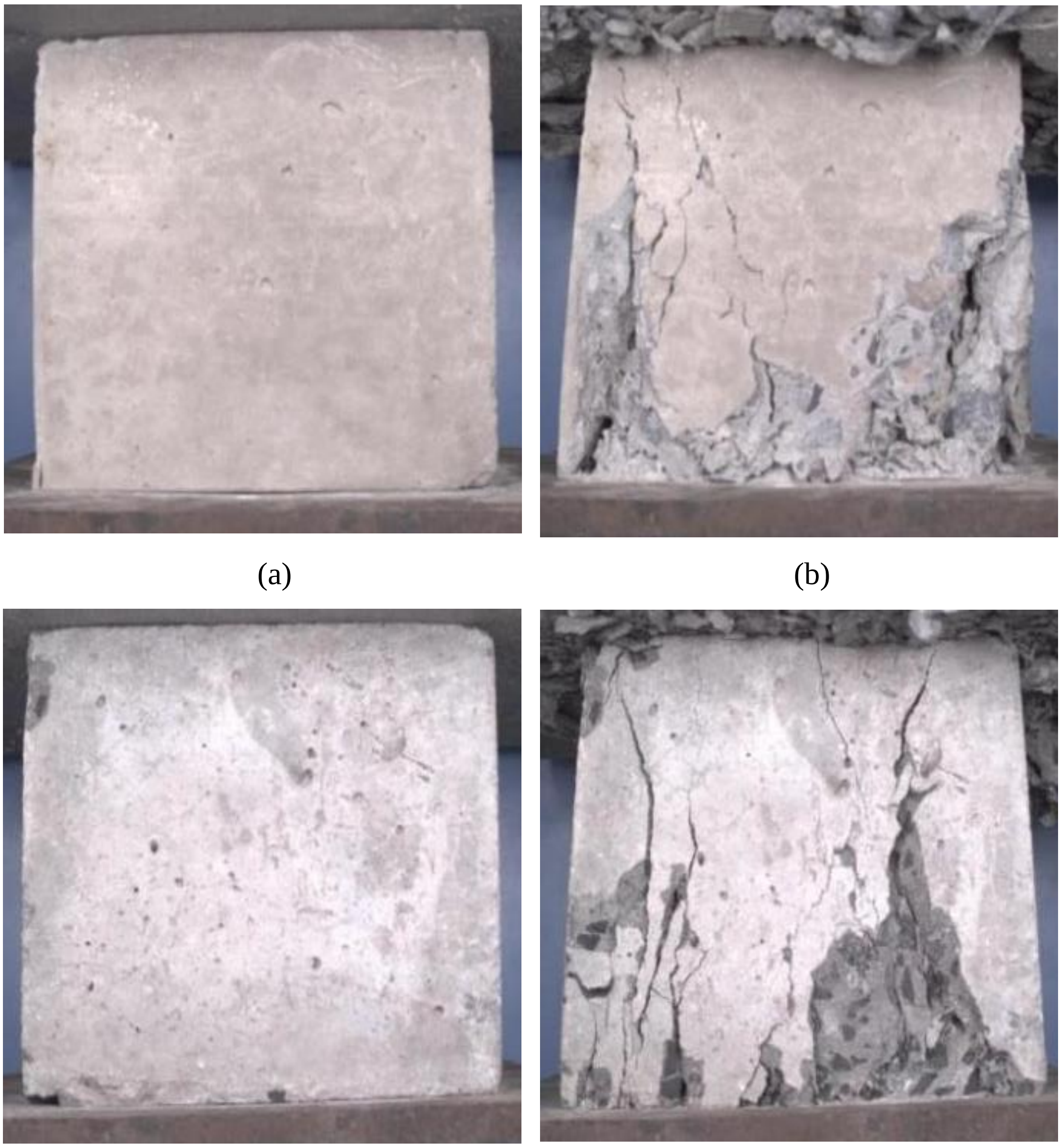
$$Area\ (in\ mm^2) = y^2 \ (15)$$

The extracted data on crack length and crack area are utilized for further monitoring and analysis. The developed software is capable of logging the concrete cube surface images, compressive load, number of cracks, crack length and crack area. The logged information can be visualized on the graphs plotted in the software. Multiple concrete cubes were tested with the developed system and observations are reported along with results.

5.3 Analysis of crack properties

Post crack detection in concrete cube surface images, the monitoring of crack occurrence and propagation with respect to the applied compressive load was carried out. In order to perform the monitoring and analysis, several tests were carried out using the developed experimental setup. Two sample images of cubes before and after compression tests acquired by the developed image acquisition setup are shown in Figure 66 (a), (b), (c), and (d). Subfigure (a) is the image of concrete cube number 1 before the test and sub-figure (b) shows the image of concrete cube number 1 after the compression test is complete. Similarly, subfigures (c) and (d) are before and after the test images of concrete cube number 2.

The images are shadowless, well oriented and have uniform illumination, which signifies the efficacy of the image acquisition setup. Image of cracked and damaged concrete cubes post-compression tests show the development of random cracks and spalling. Most of the cracks are vertical in nature and have originated from either top or bottom and propagated towards the bottom or top with the increase in applied compression load.



(a) (b)

(c) (d)

Figure 66. Sample images of cube -1 and cube -2 before and after compression test.

Data from multiple concrete cubes which includes results images, parameter graphs, and tabular data is presented in the section. The proposed machine vision system is implemented on multiple concrete cube specimens. The summary of a few compression test results in terms of load at the first crack, peak load, the total number of cracks, the total length of cracks, and the total area of cracks are presented in Table 13. Each specimen is made up of different composition.

Table 13. Summary of compression tests on concrete cubes.

Specimen	**Load at the first crack (kN)**	**Peak load (kN)**	**Total number of cracks**	**Total length of cracks (mm)**	**Total area of cracks (mm^2)**
Cube-1	55	772	7	797.99	550.16
Cube-2	150	293	9	648.87	491.12
Cube-3	310	1361	5	590.75	548.67
Cube-4	330	625	5	432.41	613.67
Cube-5	326	1351	7	486.91	510.59
Cube-6	488	1034	11	287.54	354.24
Cube-7	271	1040	8	249.83	325.96
Cube-8	495	1646	13	519.62	547.81
Average	**303.13**	**1015.25**	**8.13**	**501.74**	**492.78**

The total number of cracks shows the number of cracks which occurred during the test. The total length of cracks in millimetres (mm) shows the cumulative addition of pixel length of all the individual cracks. The total area of cracks in milli meters square (mm^2) shows the cumulative addition of pixel area of all the individual cracks detected by the developed system. The results of crack length derived in pixels and crack area in pixels square are calibrated and converted in the units of mm and mm^2 respectively. As observed in Table 13 the average value of compression load when the first crack appeared is 303.13 kN whereas the average peak load is 1015.25 kN. On average 8.13 cracks occurred on the surface of the concrete cube during the compression test. The average total length of cracks is 501.74 mm and the average total area of cracks is 492.78 mm^2. A minimum number of 3 cracks and a maximum of 13 cracks occurred on the concrete cube surface.

More detailed crack-load monitoring and analysis data are shown in Table 14. The table consists of the quantitative information on applied compression load, the time taken for the first crack to occur on the observed surface of the concrete cube, the width of the first crack, the length of the first crack, the total time of the compression test, the total number of cracks that occurred on the observed surface of concrete cube, compressive strength of cube, the total area of cracks that occurred on the observed surface of concrete cube, initial crack strength of the concrete cube, safe compressive strength of the concrete cube. Statistical information is shown in the form of the mean value of the readings and the standard deviation of the readings.

The short forms used in Table 14 are represented as No. stands for a number, L is the applied compression load in kN, CT is the time taken for the first crack occurrence in seconds, CW is the width of the first crack in mm, CL is the length of the first crack in mm, TT is the total time taken for the compression test to complete in seconds, NC is the number of cracks, CS is the compressive strength of cube in N/mm^2, AC is the cumulative area of all the cracks occurred on the surface of concrete cube measured in mm^2, ICS is the initial crack strength of cube in N/mm^2, SCS is the safe compressive strength of cube in N/mm^2, M is the mean value of all the 20 readings, and SD – is the standard deviation of all the 20 readings.

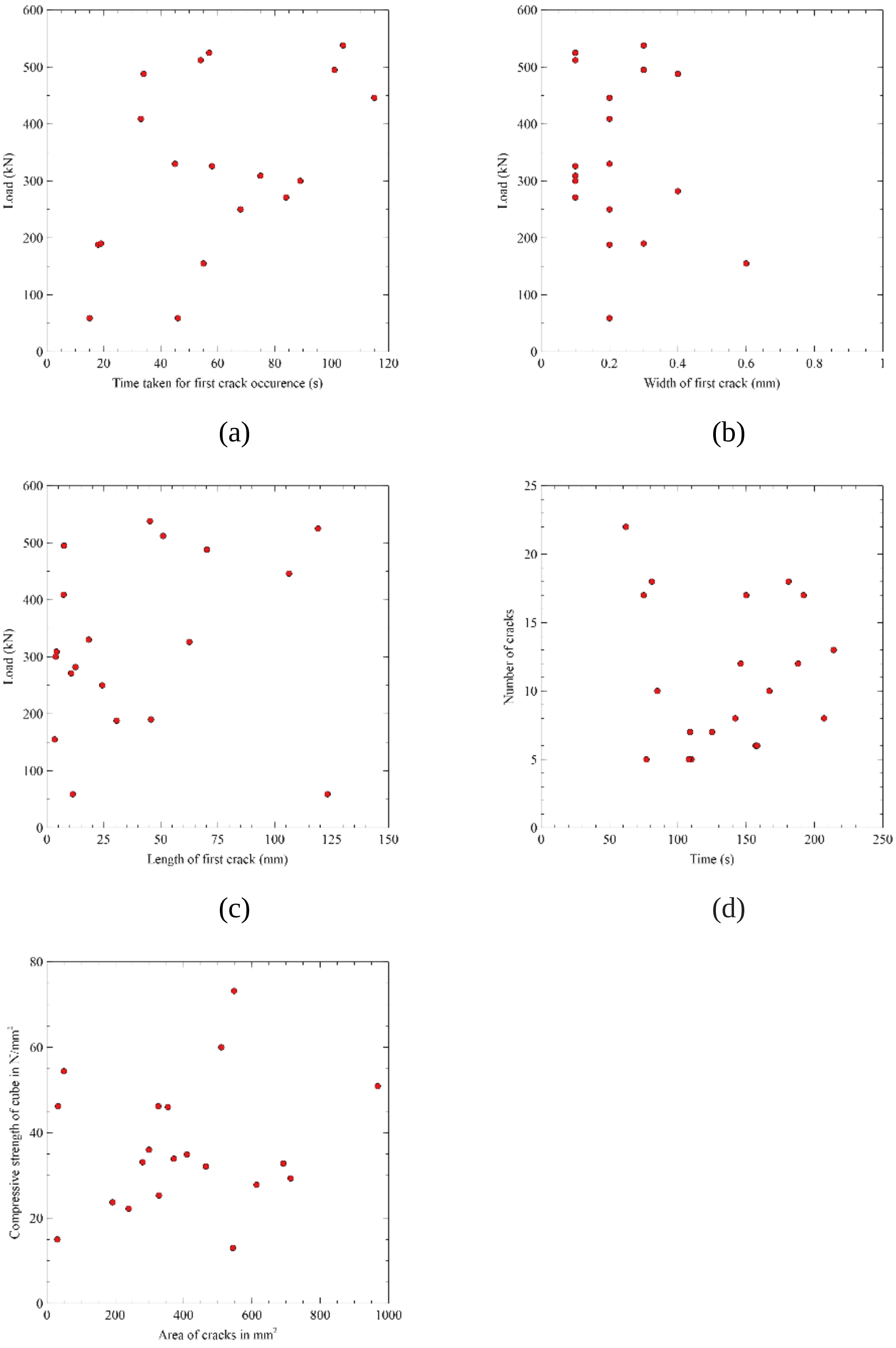
On average, the compression test time is 137 seconds while the total number of cracks occurring on the observed surface of the concrete cube is 11. The first crack occurs at an average load of 356 kN while the average time taken for the first crack to occur is 64 seconds. The average value of crack width and crack length is 0.22 mm and 38.08 mm respectively. The testing data shown in the table comprises the concrete cube of compressive strength in the range of 15 N/mm^2 to 80 N/mm^2.

The average compressive strength was 36.80 N/mm^2 and at the same time, the average area of all the cracks that occurred on the observed surface of the concrete cube at the end of the compression test is 398.39 mm^2. The average initial crack strength is 15.84 N/mm^2 and the safe compressive strength of the concrete cube is 12.27 N/mm^2. On equating the load at the first crack and the ultimate or peak load of the concrete cube, it was found that the first crack occurred at an average of 49% of the ultimate load. Similarly, the time taken for the first crack to occur is 46% of the average of total time taken for the compression test.

Table 14. Summary of crack monitoring and analysis.

No.	L	CT	CW	CL	TT	NC	CS	AC	ICS	SCS
1	59	15	0.2	11.36	125	7	33.91	371.47	2.62	11.30
2	330	45	0.2	18.37	77	5	27.78	613.67	14.67	9.26
3	326	58	0.1	62.53	110	5	60.04	510.59	14.49	20.01
4	525	57	0.1	119.04	109	7	29.33	713.38	23.33	9.78
5	512	54	0.1	51.01	108	5	33.07	280.13	22.76	11.02
6	155	55	0.6	3.41	157	6	22.22	239.30	6.89	7.41
7	250	68	0.2	24.30	142	8	23.73	192.13	11.11	7.91
8	282	125	0.4	12.54	192	17	12.98	544.40	12.53	4.33
9	300	89	0.1	3.87	146	12	15.02	30.37	13.33	5.01
10	409	33	0.2	7.34	62	22	25.33	328.02	18.18	8.44
11	538	104	0.3	45.25	214	13	35.96	298.64	23.91	11.99
12	59	46	0.2	123.19	167	10	34.89	410.00	2.62	11.63
13	997	92	0.1	4.22	158	6	46.18	32.74	44.31	15.39
14	488	34	0.4	70.24	75	17	45.96	354.24	21.69	15.32
15	188	18	0.2	30.62	81	18	50.89	968.12	8.36	16.96
16	190	19	0.3	45.69	85	10	54.40	49.74	8.44	18.13
17	446	115	0.2	106.28	188	12	32.13	465.25	19.82	10.71
18	271	84	0.1	10.63	207	8	46.22	325.96	12.04	15.41
19	309	75	0.1	4.24	150	17	32.76	691.87	13.73	10.92
20	495	101	0.3	7.52	181	18	73.16	547.81	22.00	24.39
21 (M)	**356**	**64**	**0.22**	**38.08**	**137**	**11**	**36.80**	**398.39**	**15.84**	**12.27**
22 (SD)	**211**	**33**	**0.13**	**39.51**	**47**	**5**	**15.20**	**241.97**	**9.39**	**5.07**

Figure 67 shows the graphs of crack versus load monitoring and analysis. The sub-figure (a) shows the graph of the applied compressive load in kN versus the time taken for the first visible crack to occur measured in seconds. The sub-figure (b) shows the graph of the applied compressive load in kN versus the width of the first crack measured in millimetres.



(a)

(b)

(c)

(d)

(e)

Figure 67. Crack and load monitoring graphs.

Similarly, sub-figure (c) shows the graph of the applied compressive load in kN versus the length of the first crack measured in millimetres. The sub-figure (d) shows the information on the total number of cracks versus the total time taken for the compression test in seconds. The sub-figure (e) shows the data of compressive strength of the cube measured in N/mm^2 versus the total area of all the cracks at the ultimate load measured in mm^2.

5.4 Analysis of crack-load data

Crack monitoring and analysis with respect to the applied compression load are performed on different concrete cubes. The monitoring comprises of initiation and end of each crack along with load at the crack occurrence, time, crack length, crack area and location of the crack. The remark for each crack is also mentioned. The study is carried out on multiple concrete cubes of different strengths; however, crack-load analysis of a few cubes is presented in the chapter. The observation and result images are also presented in the results. For differentiation between multiple concrete cubes, each cube has named a cube – A, cube – B, cube – C and onwards.

5.4.1 Concrete cube - A

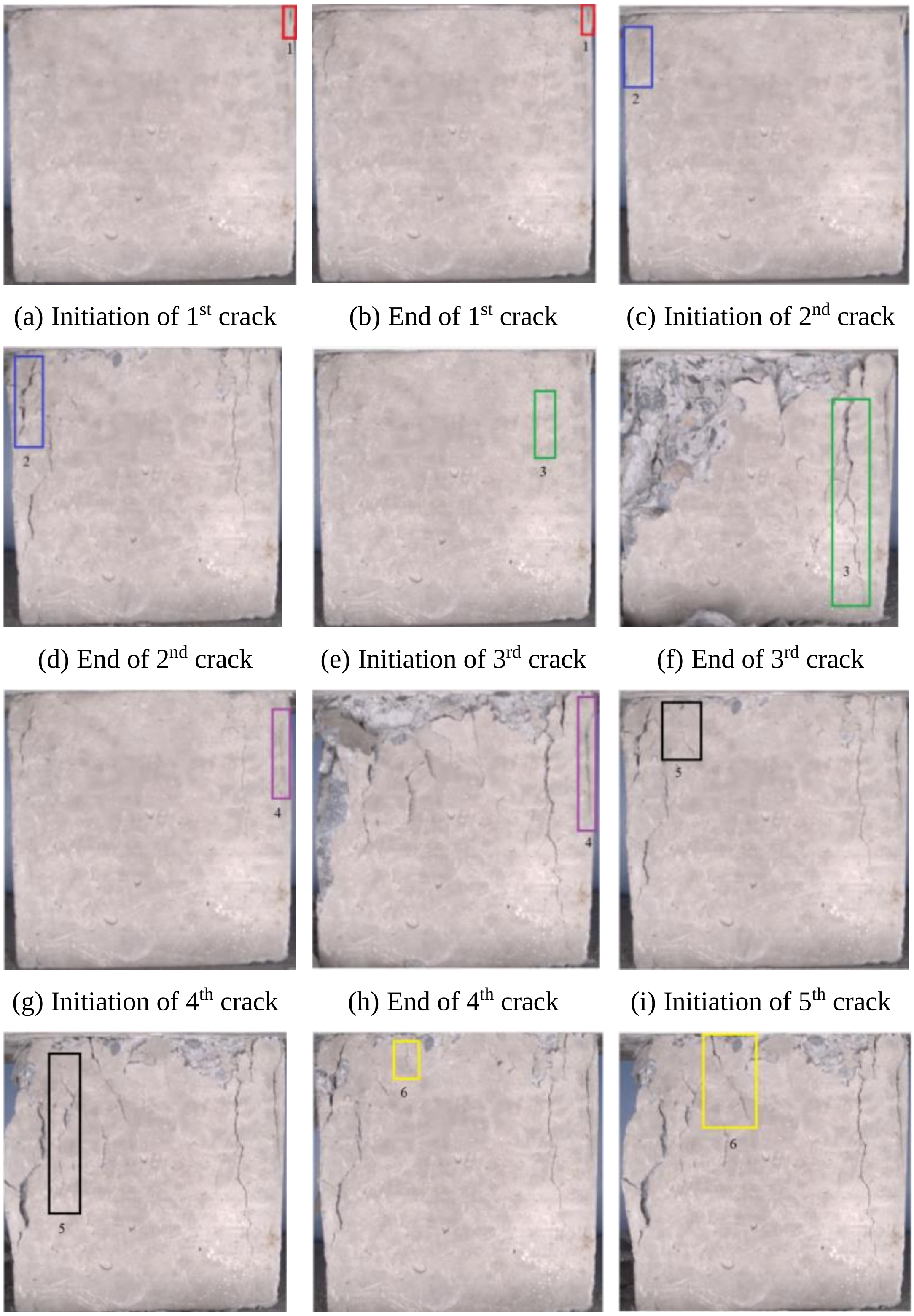
Table 15 shows the crack-load analysis of the studied concrete cube. The table shows the data of crack numbers assigned in the order of their occurrence, steps of cracking comprising of initial and final, load in terms of kN, time in seconds, crack length in mm, crack area in mm^2 and location of crack on the cube surface. The location of the crack is considered with reference to the four edges and four corners of the cube.

Table 15. Crack load analysis of a concrete cube-A.

<table>
<tr><th>Crack Number (Color)</th><th>Steps of Cracking</th><th>Load (kN)</th><th>Time (s)</th><th>Crack Length (mm)</th><th>Crack Area (mm^2)</th><th>Location</th></tr>
<tr><td>0</td><td>--</td><td>148</td><td>30</td><td>0</td><td>0</td><td>--</td></tr>
<tr><td colspan="7">Remarks: No Cracks</td></tr>
<tr><td rowspan="2">1
(Red)</td><td>Initial</td><td>155</td><td>55</td><td>3.41</td><td>2.98</td><td rowspan="2">Top Right Corner</td></tr>
<tr><td>Final</td><td>243</td><td>70</td><td>8.26</td><td>9.96</td></tr>
<tr><td colspan="7">Remarks: Visually small crack at the time of initiation and gets removed from the concrete cube with the progressive load.</td></tr>
<tr><td rowspan="2">2
(Blue)</td><td>Initial</td><td>300</td><td>75</td><td>31.18</td><td>6.72</td><td rowspan="2">Top Left Corner</td></tr>
<tr><td>Final</td><td>366</td><td>90</td><td>58.54</td><td>56.18</td></tr>
<tr><td colspan="7">Remarks: Visible crack at the time of initiation and the crack expands in length and area, which leads to spalling with the progressive load.</td></tr>
<tr><td rowspan="2">3
(Green)</td><td>Initial</td><td>340</td><td>84</td><td>31.91</td><td>13.5</td><td rowspan="2">Bottom left</td></tr>
<tr><td>Final</td><td>460</td><td>107</td><td>131.63</td><td>106.16</td></tr>
<tr><td colspan="7">Remarks: Visible Diagonal Crack at the time of initiation, which increases in the area with progressive load</td></tr>
<tr><td rowspan="2">4
(Violet)</td><td>Initial</td><td>460</td><td>107</td><td>32.30</td><td>11.99</td><td rowspan="2">Top Right</td></tr>
<tr><td>Final</td><td>480</td><td>118</td><td>67.75</td><td>59.85</td></tr>
<tr><td colspan="7">Remarks: Visible thin vertical crack which initiates from the top edge and propagates towards the bottom with an increase in length and area</td></tr>
<tr><td rowspan="2">5
(Black)</td><td>Initial</td><td>500</td><td>125</td><td>50.56</td><td>7.15</td><td rowspan="2">Right</td></tr>
<tr><td>Final</td><td>450</td><td>138</td><td>116.35</td><td>189.94</td></tr>
<tr><td colspan="7">Remarks: Visible diagonal crack at the time of initiation, which leads to spalling with progressive load</td></tr>
<tr><td rowspan="2">6
(Yellow)</td><td>Initial</td><td>463</td><td>113</td><td>28.03</td><td>9.48</td><td rowspan="2">Centre</td></tr>
<tr><td>Final</td><td>395</td><td>157</td><td>53.99</td><td>31.09</td></tr>
<tr><td colspan="7">Remarks: Visually thin diagonal crack at the centre of the cube, which increases in length and area as the load propagates</td></tr>
</table>

No crack appeared on the concrete cube until 148 kN of compressive load. The first crack was initiated at 155kN with a 3.41 mm crack length and 2.98 mm^2 crack area. Crack number 1 is removed from the concrete surface due to spalling of the concrete. The final load of 243 kN along with an 8.26 mm long crack and 9.96 mm^2 cracks was reported. Crack numbers 2 and 3 occurred with progression in compressive load at 300 kN and 340 kN respectively. Crack number 2 was visually thin at the initiation. However, the crack expanded from 6.72 mm^2 to 56.18 mm^2 crack area. Crack number 3 occurred in the right half of the cube and propagated vertically toward the bottom edge of the cube. Length of the crack number 3 was increased from 31.91 mm to 131.63 mm with respect to load. At 460 kN load, the fourth crack occurred quickly after the third crack had terminated.

Initially, the crack was visually thin and vertical which propagated from top edge to bottom edge till 480 kN load. The fifth crack occurred after the peak load of the cube and ended at 450 kN. The crack was diagonal in nature which was removed due to spalling of concrete. Crack number 6 appeared at a load of 463 kN and stopped at a load of 395 kN. Individual images of each crack initiation and termination are shown in Figure 68. Each crack is highlighted with a bounding box and numbered appropriately to match the crack-load analysis reported in Table 15.



(a) Initiation of 1st crack (b) End of 1st crack (c) Initiation of 2nd crack

(d) End of 2nd crack (e) Initiation of 3rd crack (f) End of 3rd crack

(g) Initiation of 4th crack (h) End of 4th crack (i) Initiation of 5th crack

(j) End of 5th crack (k) Initiation of 6th crack (l) End of 6th crack

Figure 68. Concrete cube-A images showing the initial crack and final crack.

Figure 69 (a) represents the graph of crack length in mm vs load in kN of concrete cube-A while sub-figure (b) shows the graph of the crack area in mm^2 versus load in kN. Each of the cracks is shown with different colours and markers in order to differentiate from other cracks. The crack length and crack area progression with respect to applied load is shown for all the six cracks occurred during the compression test of concrete cube – A. The length of crack numbers 1,2 and 3 is increasing with a load while the length of crack 4 increases first and then decreases. This is because the peak load capacity of the concrete cube – A has reached and thereafter the load starts decreasing. However, the length of crack number 4 increases with a decrease in the load. The length of crack numbers 5 and 6 is linearly increasing with respect to a decrease in the applied load. Similar patterns are obtained in the graphs of load versus crack area. The area of crack numbers 1,2, and 3 are increasing with the increase in load. While the area of crack number 4 increases initially, but decreases thereafter due to peak load. Crack numbers 5 and 6 shows an increase in crack area with respect to a decrease in load.

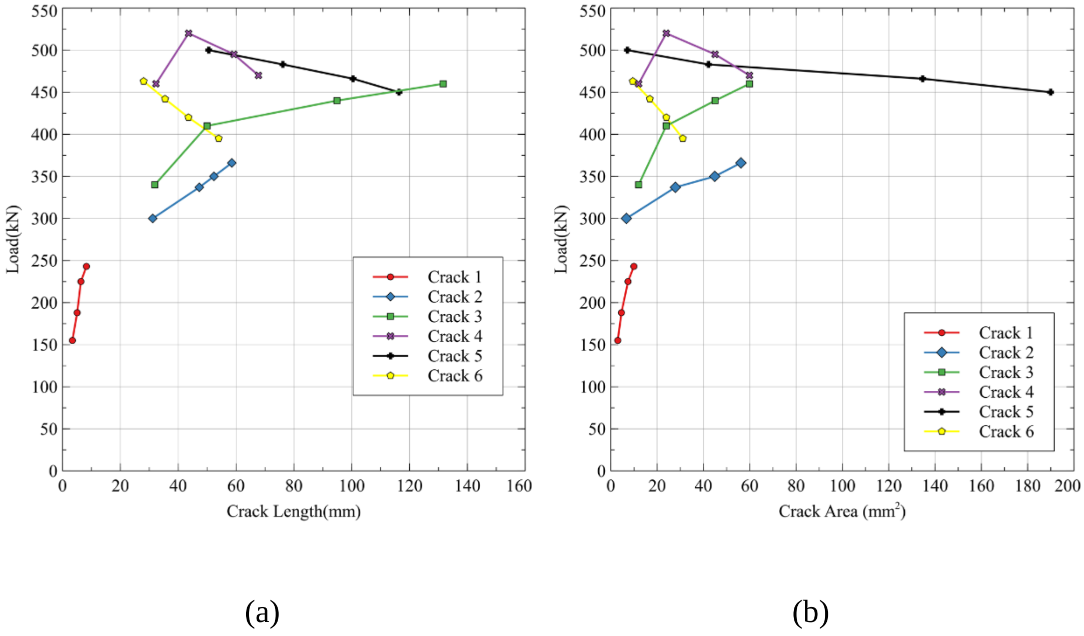


Figure 69. Progressive and individual crack monitoring with applied load for cube-A.

The graphs reported in Figure 69 depict the monitoring and analysis capability of the methodology. The propagation of individual cracks in terms of length and area can be studied with respect to the applied load.

Table 16, Table 17, Table 18, and Table 19 cumulatively showcase the progression of all six cracks with reference to load, crack length and crack area. Four data points of each crack are presented for analysis. The crack length and crack area of the first crack initiation show that the crack is smaller both in terms of length and area whereas the crack initiation length is longer in comparison with the area. It shows that crack number 2 is longer in length and smaller in area. However, with an increase in applied load, the length of the second is increased by 1.89 times the initial length and the crack area is increased by 8.3 times the initial crack area. In the case of crack number 1, the length is increased by 1.86 times the initial length and the crack area is increased by 1.36 times the initial area.

Table 16. Progression details of crack numbers 1 and 2 in concrete cube – A.

Load(kN)	**Length of 1st crack (mm)**	**Area of 1st crack (mm^2)**	**Load(kN)**	**Length of 2nd crack (mm)**	**Area of 2nd crack (mm^2)**
155	3.41	2.98	300	31.18	6.72
188	4.06	3.71	337	47.30	27.88
225	4.72	3.94	350	52.36	44.88
243	6.35	4.07	366	58.54	56.18

The progression of crack numbers 3 and 4 with respect to applied load in terms of crack length and area are shown in Table 17.

Table 17. Progression details of crack numbers 3 and 4 in concrete cube – A.

Load(kN)	**Length of 3rd crack (mm)**	**Area of 3rd crack (mm^2)**	**Load(kN)**	**Length of 4th crack (mm)**	**Area of 4th crack (mm^2)**
340	31.91	13.50	460	32.30	11.99
420	50.00	44.74	520	43.60	23.92
440	94.89	83.32	495	59.27	45.02
480	131.63	106.16	470	67.75	59.85

A large increase in the length and area of crack number three is observed during the progression with respect to load. An increase of 4.12 times and 7.86 times is observed in crack length and crack area respectively. The increase is calculated with respect to the initial crack length and area. Similarly, an increase of 2.09 times is observed in crack length and 4.99 times increase in crack area is observed for crack number 4.

Table 18 consists of the progression data of crack numbers 5 and 6. The cracks are detected post the peak load of the concrete cube. Due to this, the length and area are increasing but the load is decreasing. An increase of 2.30 times and 26.56 times are observed in crack length and area of crack number 5 respectively. On the other hand, an increase of 1.92 times and 3.27 times are obtained in crack length and crack area of crack number 6 respectively.

Table 18. Progression details of crack numbers 5 and 6 in concrete cube – A.

Load(kN)	**Length of 5th crack (mm)**	**Area of 5th crack (mm^2)**	**Load(kN)**	**Length of 6th crack (mm)**	**Area of 6th crack (mm^2)**
500	50.56	7.15	463	28.03	9.48
483	76.18	42.17	442	35.45	16.84
466	100.45	134.67	420	43.54	23.95
450	116.35	189.94	395	53.99	31.09

From the results and analysis discussed above, it can be concluded that the area of crack number 5 has shown highest increment whereas increase in the length of crack number 3 is highest as compared with the other cracks. Table 19 depicts the quantitative data of load and crack monitoring. The table consists of time, load, length of crack, and area of the crack at the end of the crack along with the crack number and cumulative data of length and area. The cumulative length of cracks is obtained by adding the length of previous crack with the length of current crack. Similarly, the cumulative crack area is also calculated. Considering the crack length as 150 mm, the cumulative length of all 6 cracks is 436.52 mm which is around 3 times the length of the concrete cube. Similarly, the cumulative area of all the cracks is 453.18 mm^2 which is around 2% of the total area of the concrete cube surface which is considered 22,500 mm^2.

Table 19. Load versus crack monitoring data for concrete cube-A.

Time (sec)	**Load (kN)**	**Crack number**	**Length of crack at the end (mm)**	**Area of crack at the end (mm^2)**	**Cumulative length of cracks (mm)**	**Cumulative area of cracks (mm^2)**
30	148	0	0	0	0	0
70	243	1	8.26	9.96	8.26	9.96
90	366	2	58.54	56.18	66.80	66.14
107	460	3	131.63	106.16	198.43	172.30
118	480	4	67.75	59.85	266.18	232.15
138	450	5	116.35	189.94	382.53	422.09
157	395	6	53.99	31.09	436.52	453.18

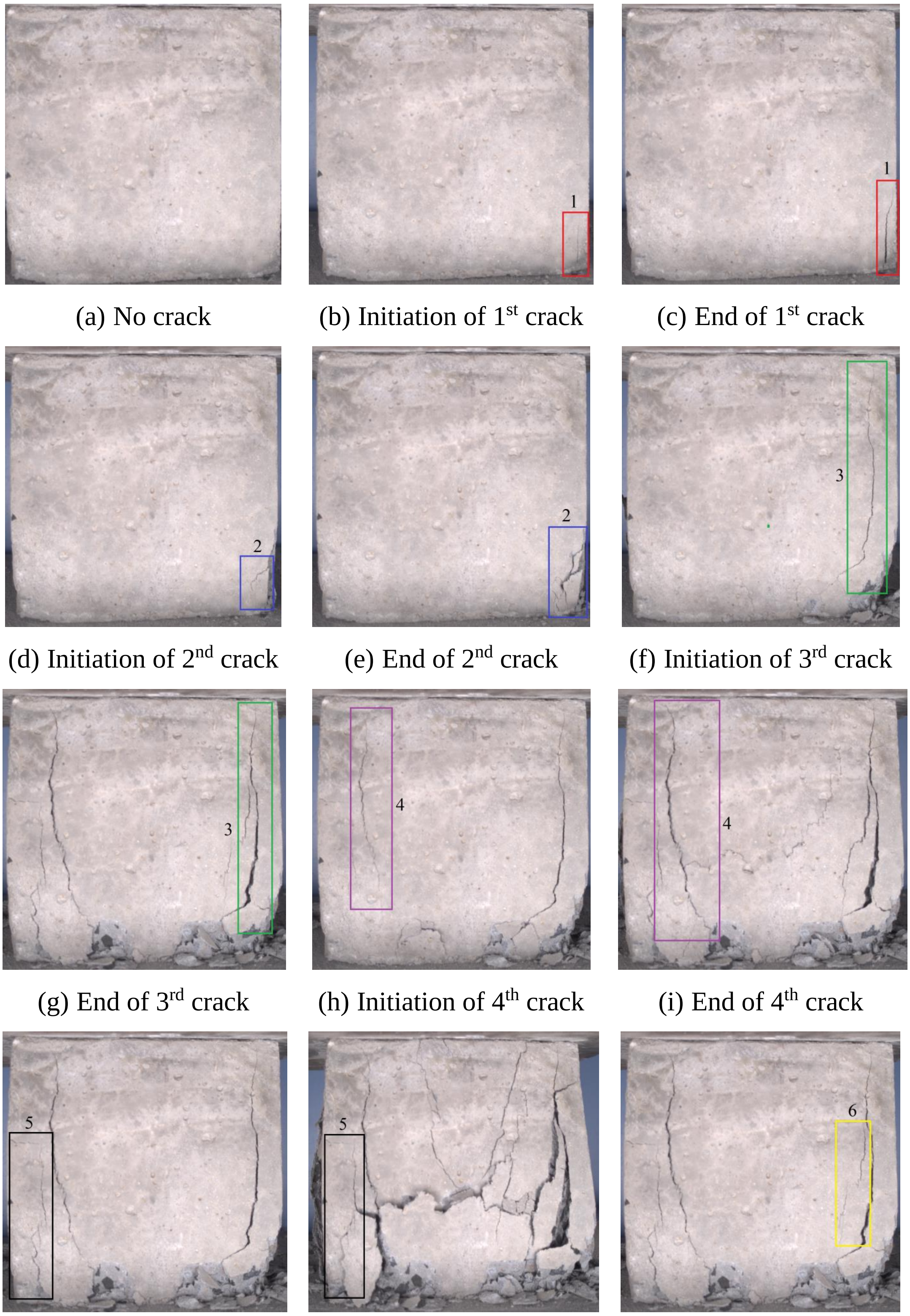
5.4.2 Concrete cube - B

Concrete cube sample-B crack load analysis data is discussed. Quantitative data of crack number, load in kN, time in seconds, individual crack length in mm at initiation and ending, and individual crack area in mm^2 measured at initiation and ending, the location of the crack, and remarks of the crack are mentioned collectively in Table 20. The tabulated data of individual cracks are related to the images shown in Figure 70 (a) - (l). The crack location is considered in relation to the four edges as well as the four corners of the cube. The obtained results for the cube sample show that no crack occurs till 55 kN compressive load and 45 seconds. The first crack initiates at 59 kN load with a crack length of 11.36 mm and 1.54 mm^2 crack area, while it ended at 164 kN at 60 seconds with a crack length of 43.56 mm and a crack area of 23.61 mm^2. The crack occurred near the bottom right corner and progressed in a diagonal direction towards the bottom edge. Crack number 1 disappears with progression in load value due to spalling of concrete. Similarly, crack numbers 2 to 7 initial load, time, crack length, crack area, location and remarks are presented in the table.

Table 20. Crack load analysis of a concrete cube-B.

Crack Number (Color)	**Steps of Cracking**	**Load (kN)**	**Time (s)**	**Crack Length (mm)**	**Crack Area (mm^2)**	**Location**
0	--	55	45	0	0	--
Remarks: No Cracks						
1 (Red)	Initial	59	46	11.36	1.54	Bottom right corner
	Final	164	60	43.56	23.62	
Remarks: Visible diagonal crack at the time of initiation and leads to spalling of concrete with an increase in load.						
2 (Blue)	Initial	310	75	19.17	7.24	Bottom right
	Final	475	90	49.76	35.02	
Remarks: Visible diagonal crack at the time of initiation which occurs along with crack number 1 and leads towards spalling of concrete with progressive load.						
3 (Green)	Initial	649	111	121.70	73.93	Top Right
	Final	757	135	130.11	90.04	
Remarks: Visually thin vertical crack which initiates from the top edge and propagates towards the bottom along with an increase in length, width and area.						
4 (Violet)	Initial	720	120	72.23	22.59	Top Left
	Final	772	140	143.51	99.52	
Remarks: Visually thin vertical crack which initiates from the top edge and propagates towards the bottom along with an increase in length, width and area.						
5 (Black)	Initial	750	130	79.33	30.72	Bottom Left
	Final	717	146	120.66	96.18	
Remarks: Visible diagonal crack at the time of initiation, which leads to spalling with progressive load						
6 (Yellow)	Initial	763	134	69.92	26.55	Right Centre
	Final	750	142	84.04	58.39	
Remarks: Visually thin and zig-zag crack at the time of occurrence, which propagates and gets connected to crack number 4.						
7 (Orange)	Initial	642	144	127.83	51.57	Centre
	Final	660	154	226.35	147.39	
Remarks: Visually thin and zig-zag vertical crack at the time of occurrence, which initiates from the top edge and propagates towards the bottom edge with expansion.						

Figure 70 (a) - (o) show the total 15 progressive images of the concrete cube-B for which crack monitoring results are presented and discussed. Images at the initiation of the crack and end of the same crack are shown in Figure 70 for a total of 7 cracks observed during the compression testing of the concrete cube-B.



(a) No crack

(b) Initiation of 1st crack

(c) End of 1st crack

(d) Initiation of 2nd crack

(e) End of 2nd crack

(f) Initiation of 3rd crack

(g) End of 3rd crack

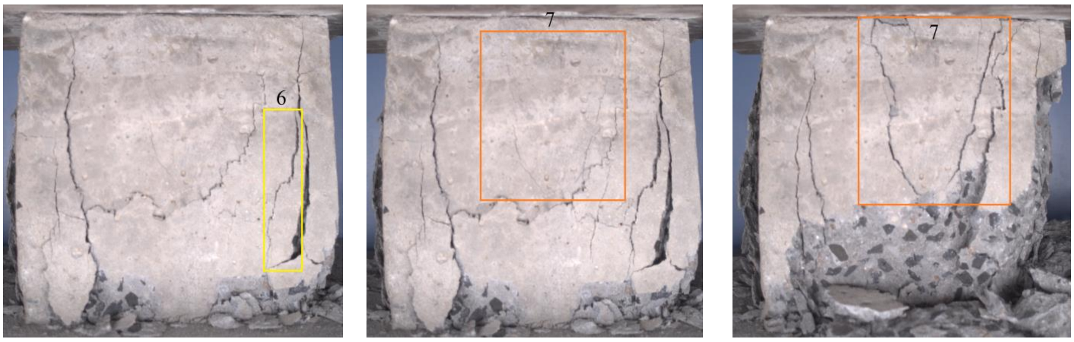
(h) Initiation of 4th crack

(i) End of 4th crack

(j) Initiation of 5th crack

(k) End of 5th crack

(l) Initiation of 6th crack



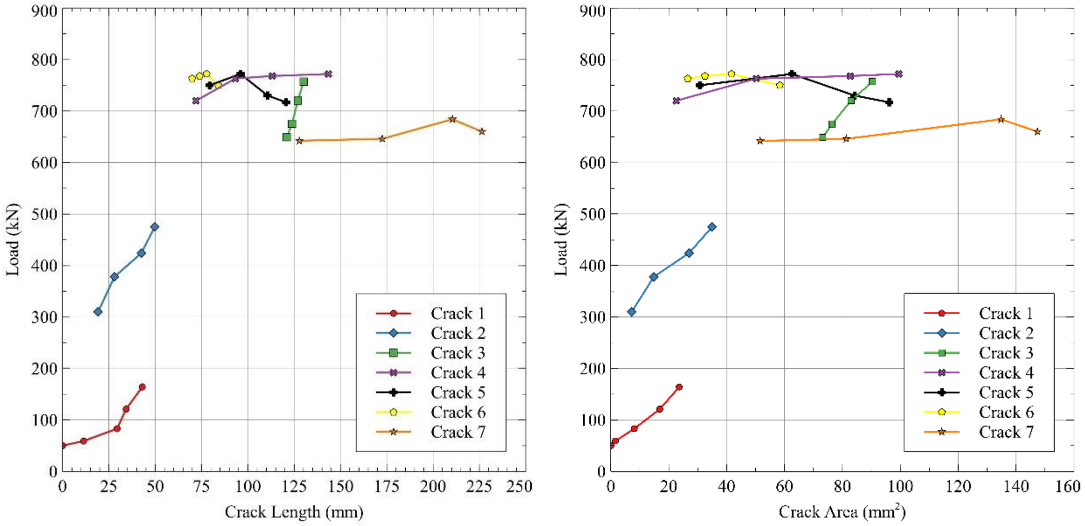
(m) End of 6th crack (n) Initiation of 7th crack (o) End of 7th crack

Figure 70. Concrete cube-B images showing initial crack and final crack.

Careful observation of Figure 70 (b)- (f) depicts that the spalling of concrete occurred and concrete material is removed near crack number 1 and 2. Each crack is marked with a different colour bounding box to understand and interpret its progression. As mentioned in the remarks, the majority of the cracks are vertical, initiate from the top edge and propagate towards the bottom edge. The progression of each crack in terms of length and area can be visualized as well as interpreted from the results. At the time of occurrence, cracks are visually very thin and difficult to detect and trace with the naked eye. However, with an increase in the compression load, the cracks expand both in horizontal as well as vertical directions.

The integrated system comprising of machine vision and CNN developed in the present work not only enables crack detection in real-time but also provides the crack propagation data with respect to the applied load. The quantitative results obtained for individual cracks with respect to load and time can provide important information about the behaviour of the concrete cube under compressive loading.

Figure 71 (a) shows the progression of individual cracks in terms of crack length plotted versus the applied load for concrete cube B. The graphical results are related to images shown in Figure 70 (a) - (o) and quantitative results shown in Table 20. The length of crack numbers 1 to 7 is shown to understand the progression of individual cracks. The majority of the cracks occur near peak load and instantly appear on the monitored surface due to which crack numbers 2 to 7 are not initiated with nearly zero crack length. Due to the computation load of a CNN, image and load data are acquired at a rate of 1 sample per second. The crack length of individual cracks may surpass the total length of the cube which is 150 mm due to the zig-zag or diagonal nature of cracks. The length of the majority of cracks increases with an increase in the applied load but near the peak load, crack length may suddenly increase as observed from the graph of crack number 7.



(a) (b)

Figure 71. Progressive and individual crack monitoring with applied load for cube-B.

Figure 71 (b) shows the progression of individual cracks in terms of crack area in mm^2 versus applied compressive load in kN. The area of crack numbers 1 to 7 is shown to understand the progression of an individual crack. Although the crack length may overshoot the length of the cube which is 150 mm, the area of the crack will never overshoot the total area of the cube which is 22,500 mm^2. From the graphs, it is evident that the maximum crack length is 226.35 mm while the maximum crack area is 147.39 mm^2. The observation can be interpreted from the image shown in Figure 70 (a) - (0) which depicts that the crack propagates and expand in a vertical direction rather than a horizontal direction.

Table 21, Table 22, Table 23 and Table 24 cumulatively showcase the progression of all seven cracks with reference to load, crack length and crack area. The progression is shown with four data points for each parameter. The increase in the area with respect to the increase in load for crack numbers 1 and 2 is higher as compared to the increase in the length. An increase of 4.83 times and 15.33 times is observed in the crack area of crack numbers 1 and 2 respectively. On the other hand, an increase of 3.83 times and 2.59 times in crack length is observed for crack numbers 1 and 2 respectively.

Table 21. Progression details of crack numbers 1 and 2 in concrete cube – B.

Load(kN)	**Length of 1st crack (mm)**	**Area of 1st crack (mm^2)**	**Load(kN)**	**Length of 2nd crack (mm)**	**Area of 2nd crack (mm^2)**
59	11.36	1.54	310	19.17	7.24
83	29.43	8.10	378	27.99	14.84
121	34.31	16.91	424	42.52	26.97
164	43.56	23.62	475	49.76	35.02

The progression of crack numbers 3 and 4 with respect to applied load in terms of crack length and area is shown in Table 22. A large increase in the length and area of crack number three is observed during the progression with respect to load. An increase of 1.08 times is observed in the length of crack number 3 whereas a 1.23 times increase is observed in the area of the crack. Similarly, an increase of 1.99 times is observed in crack length and 4.40 times increase in crack area is observed for crack number 4.

Table 22. Progression details of crack numbers 3 and 4 in concrete cube – B.

Load(kN)	**Length of 3rd crack (mm)**	**Area of 3rd crack (mm^2)**	**Load(kN)**	**Length of 4th crack (mm)**	**Area of 4th crack (mm^2)**
649	121.00	73.23	720	72.00	22.59
675	123.84	76.45	763	93.24	50.26
720	126.96	82.96	768	113.19	82.71
757	130.11	90.40	772	143.51	99.52

Table 23 consists of the progression data of crack numbers 5 and 6. The cracks are detected post the peak load of the concrete cube. Due to this, the length and area are increasing but the load is decreasing. An increase of 1.52 times and 3.13 times are observed in crack length and area of crack number 5 respectively. On the other hand, an increase of 1.20 times and 2.19 times are obtained in crack length and crack area of crack number 6 respectively. The increase in crack length, crack area and load are smaller in magnitude as compared to that of other cracks as the peak load has reached.

Table 23. Progression details of crack numbers 5 and 6 in concrete cube – B.

Load(kN)	**Length of 5th crack (mm)**	**Area of 5th crack (mm^2)**	**Load(kN)**	**Length of 6th crack (mm)**	**Area of 6th crack (mm^2)**
750	79.34	30.72	763	69.92	26.55
772	96.06	62.60	768	74.00	32.51
730	110.64	84.24	772	77.72	41.68
717	120.66	96.18	750	84.04	58.39

Table 24 shows the crack length and crack area measurement with the applied compressive load. The crack is observed post-peak load and hence the increase in load is only 18 kN but the increase in the crack length and crack area is 98.52 and 31.04 respectively. From the results and analysis discussed above, it can be concluded that the area of crack number 2 has shown the highest increment whereas an increase in the length of crack number 1 is the highest as compared with the other cracks.

Table 24. Progression details of crack number 7 in concrete cube – B.

Load(kN)	**Length of 7th crack (mm)**	**Area of 7th crack (mm^2)**
642	127.83	16.70
646	172.60	26.35
684	210.49	43.67
660	226.35	47.74

Quantitative data presented in terms of time in seconds, load in kN, the number assigned to the crack, length of the crack in mm measured at the end, area of the crack in mm^2 measured at end of each crack, the total length of cracks in mm and total area of cracks in mm^2 are presented in Table 25.

Table 25. Load versus crack monitoring data for cube-B.

Time (sec)	**Load (kN)**	**Crack number**	**Length of crack at the end (mm)**	**Area of crack at the end (mm^2)**	**Cumulative length of cracks (mm)**	**Cumulative area of cracks (mm^2)**
45	55	0	0	0	0	0
60	164	1	43.56	23.62	43.56	23.62
90	475	2	49.76	35.02	93.32	58.64
135	757	3	130.11	90.04	223.43	148.68
120	772	4	143.51	99.52	366.94	248.20
146	717	5	120.66	96.18	487.60	344.38
142	750	6	84.04	58.39	571.64	402.77
154	660	7	226.35	147.39	797.99	550.16

The total length of all the cracks after the end of the compression test is roughly 5.31 times the length of a concrete cube whereas the total crack area is 0.24 % of the total exposed area of the concrete cube. The total length of the cube is considered 150 mm and the total area of the concrete cube surface is considered 22,500 mm^2. Data presented in Table 25 can also be used to find the rate of loading. With the use of the monitoring and analysis presented for the concrete cube – B, insightful statistics of concrete cube behavior when subjected to compression tests are observed. Similarly, analysis data of other concrete cubes are presented in upcoming sections.

5.4.3 Concrete cube – C

Table 26 shows the crack-load analysis of the studied concrete cube - C. The table comprises the data of crack numbers assigned in the order of their occurrence along with colour code, steps of cracking comprising of initial and final data, load in terms of kN, time in seconds, crack length in mm, crack area in mm^2 and location of crack on the cube surface. Also suitable for remarks related to crack size, propagation direction and starting edge for each crack is mentioned.

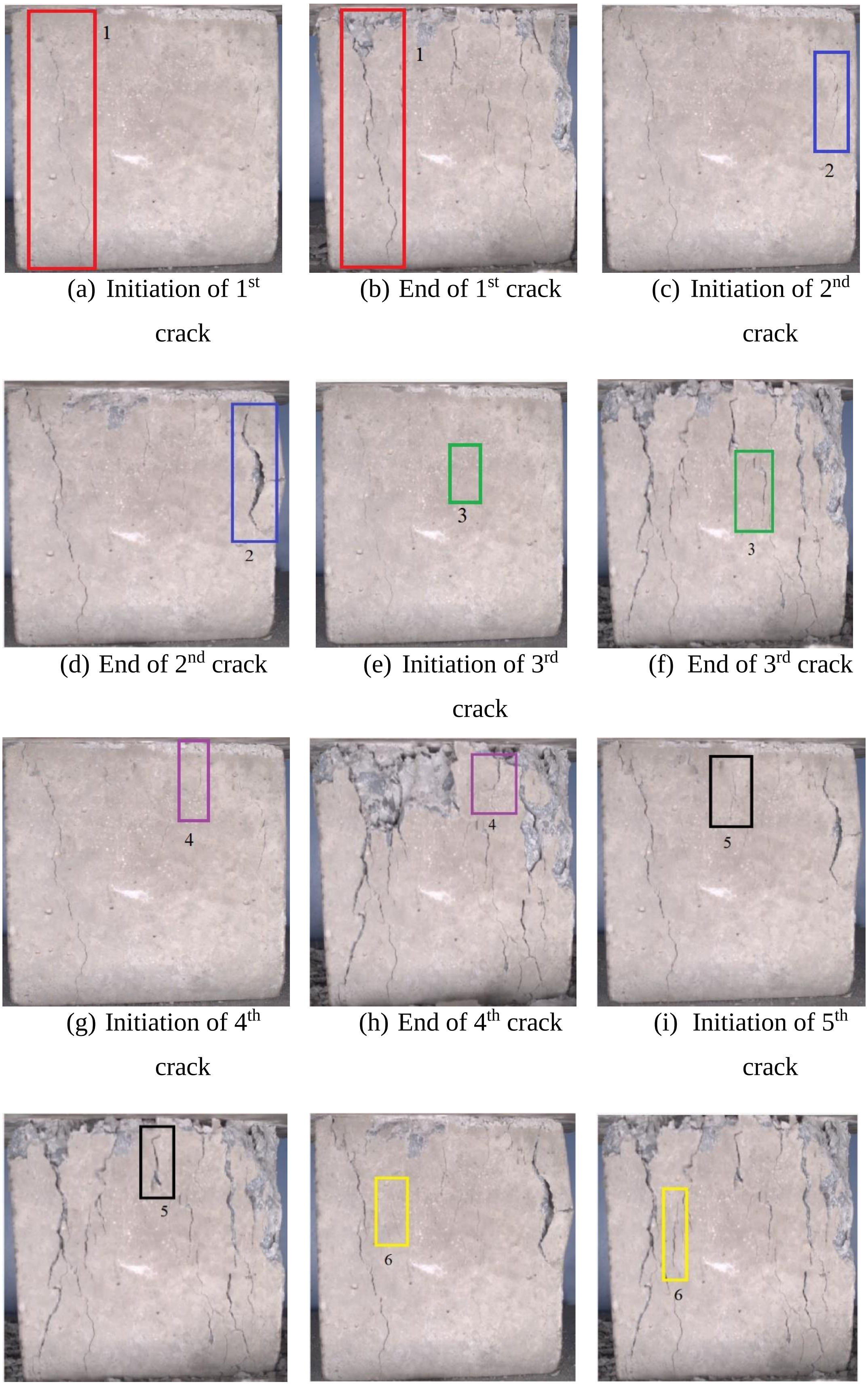
Table 26. Crack load analysis of a concrete cube – C.

Number of Cracks (Colour)	**Steps of Cracking**	**Load (kN)**	**Time (s)**	**Crack Length (mm)**	**Crack Area (mm^2)**	**Location**
0	---	280	35	---	---	---
1 (Red)	Initial	284	41	25.91	8.37	Left
	Final	297	66	103.90	102.94	
Remark: Visually thin vertical crack which initiates from the top edge and propagates to the bottom along with an increase in length, width and area.						
2 (Blue)	Initial	284	50	46.42	15.77	Right
	Final	292	73	72.46	44.45	
Remark: Visible thin vertical crack at the time of initiation leads to spalling of concrete with an increase in load.						
3 (Green)	Initial	289	54	55.06	15.36	Centre
	Final	267	80	65.45	28.87	
Remark: Visible vertical crack at the time of initiation, whose length and area increase with progressive load.						
4 (Violet)	Initial	290	60	23.94	7.85	Top Right
	Final	263	75	32.00	19.38	
Remark: Visible vertical crack at the time of initiation, whose length and area increase with progressive load.						
5 (Black)	Initial	297	63	24.32	9.04	Centre
	Final	258	88	47.09	39.13	
Remark: Visually thin vertical crack which initiates from the top edge and propagates to the bottom along with an increase in length, width and area.						
6 (Yellow)	Initial	296	65	22.29	8.32	Left
	Final	195	92	54.26	36.43	
Remark: Visible vertical crack at the time of initiation, whose length and area increase with progressive load.						
7 (Orange)	Initial	291	75	41.56	14.73	Right
	Final	232	100	98.25	57.92	
Remark: Visually thin vertical crack which initiates from the top edge and propagates to the bottom along with an increase in length, width and area.						
8 (Pink)	Initial	288	80	29.18	8.83	Right
	Final	241	102	61.77	28.08	
Remark: Visible vertical crack at the time of initiation, whose length and area increase with progressive load.						
9 (Grey)	Initial	270	83	62.37	26.33	Left
	Final	155	110	113.81	133.92	
Remark: Visually thin vertical crack which initiates from top edge and propagates to the bottom along with an increase in length, width and area.						

A total of 9 cracks are observed in the analysis. The majority of the cracks occurred during the peak load of the concrete cube. Till the load of 280 kN which is measured at 35 seconds from the start of the compression test, no crack appeared on the concrete cube surface. The first crack was initiated at 284kN with a 25.91 mm crack length and 8.37 mm^2 crack area. The crack started from the top edge of the cube and grew in a vertical direction till the bottom edge of the cube. The length and area of crack number 1 are 103.90 mm and 102.94 mm^2 at the end with a load of 297 kN.

Crack numbers 2 and 3 occurred with progression in compressive load simultaneously with crack number 1 at 284 kN and 289 kN respectively. Crack number 2 was visually thin at the initiation with an increase in load, the crack is removed due to spalling of concrete. Length of the crack number 3 was increased from 55.06 mm to 65.45 mm with respect to load. Crack numbers 4,5,6,7,8,9 started at 290 kN, 297 kN, 296 kN, 291 kN, 288 kN, and 270 kN load respectively. All these cracks occurred during the peak load and have grown in a vertical direction propagating in the downward direction. Individual images of each crack initiation and terminated are shown in Figure 72. Each crack is highlighted with a bounding box and numbered appropriately to match the crack-load analysis reported in Table 26.

Crack-load analysis carried out for other cube-C using the machine vision system developed is shown in Figure 72. Cube-C had a total of 9 cracks during the compression test. The progression of each crack is shown with a different coloured line and line marker. The cracks which occurred on the surface during the initial loading of the test show increase in the crack length and area with respect to the applied load. The increase continues till the end of the compression test wherein the concrete cube will completely fail. However, a few cracks also occur post the peak load and the progression of these cracks is observed while the load-carrying capacity of the cube is decreasing. The length and area of such cracks show a downward trend with respect to the applied load.



(a) Initiation of 1st crack

(b) End of 1st crack

(c) Initiation of 2nd crack

(d) End of 2nd crack

(e) Initiation of 3rd crack

(f) End of 3rd crack

(g) Initiation of 4th crack

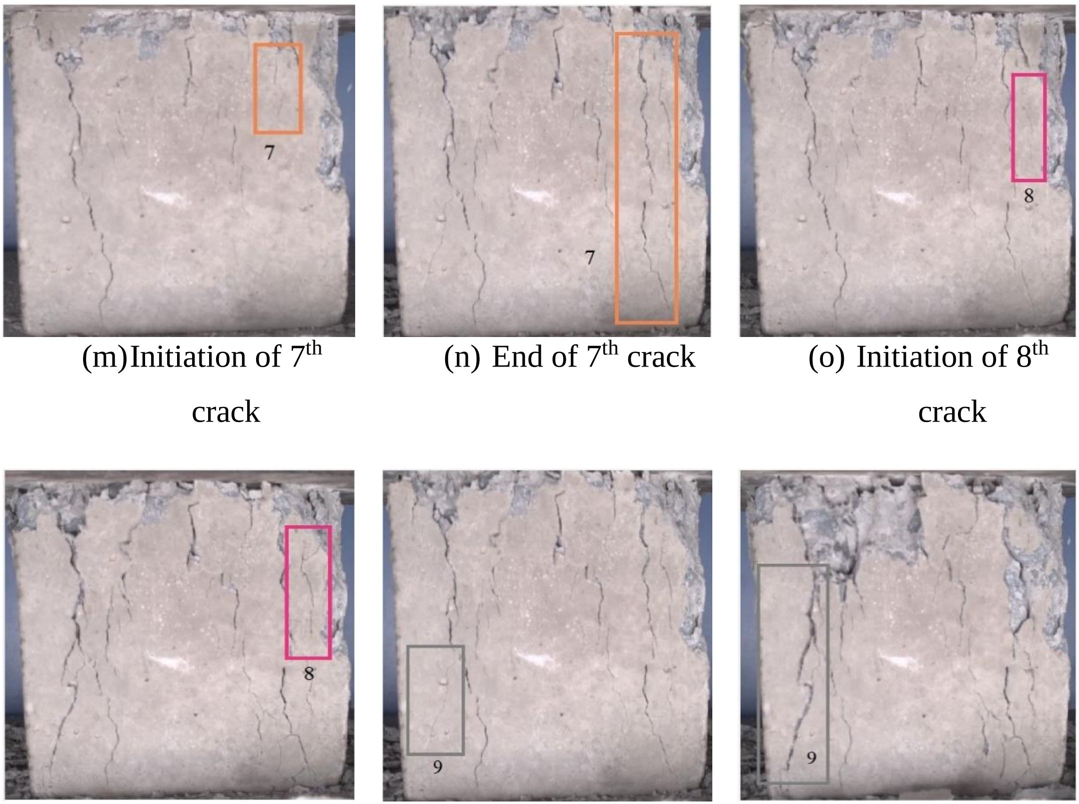
(h) End of 4th crack

(i) Initiation of 5th crack

(j) End of 5th crack

(k) Initiation of 6th crack

(l) End of 6th crack



(m) Initiation of 7th crack

(n) End of 7th crack

(o) Initiation of 8th crack

(p) End of 8th crack

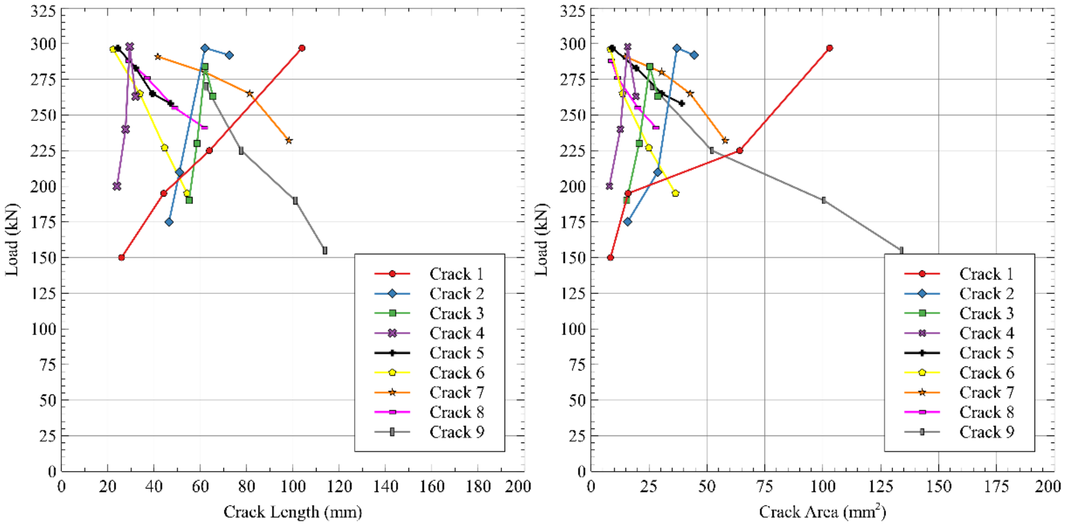
(q) Initiation of 9th crack

(r) End of 9th crack

Figure 72. Concrete cube-C images showing initial crack and final crack.

The start and end of each crack are observed in the images of the concrete cube shown in Figure 72 (a) – (r). The crack-load analysis presented in Table 26 in terms of a crack starting location, propagation direction, and crack width is interpreted from these figures of concrete cube. Initially, the majority of the cracks detected on the concrete cube surface are visually thin. With an increase in the applied compression load, the cracks expand in vertical downward directions which is evident from the results shown in Table 26 and Figure 72. The spalling of concrete is also observed in images of a few cracks. The analysed data of load and crack for individual cracks can provide vital information about the behaviour of the concrete cube under compressive loading.

Figure 73(a) represents the graph of individual crack length in mm vs load in kN of concrete cube-C and sub-figure (b) shows the graph of an individual crack area in mm^2 versus load in kN. A total of four data points for each crack are shown in the graphs. Each of the cracks is shown with different colours and markers in order to differentiate from other cracks. The crack length and crack area progression with respect to applied load is shown for all the nine cracks that occurred during the compression test of the concrete cube – C. The length and area of the first crack are increasing with respect to the increase in load. However, the length and area of crack numbers 2,3 and 4 are increasing initially with an increase in load but towards the end, they are increasing with a decrease in load. It shows that the peak load has reached when the concrete cube – C is subjected to a compression test and thereafter the load decreases. Crack numbers 6,7,8 and 9 show downward trends in length and area because the load is decreasing with an increase in length and area. Hence, it is observed that crack number 1 occurred before the peak load, crack number 2,3, and 4 occurred during the peak load and the remaining cracks 6,7,8, and 9 occurred after the peak load.



(a) Individual crack length versus load (b) Individual crack area versus load

Figure 73. Progressive and individual crack monitoring with applied load for cube-C.

Table 27, Table 28, Table 29, Table 30, and Table 31 cumulatively showcase the progression of all seven cracks with reference to load, crack length and crack area. The progression is shown with four data points for each parameter. The area of crack number 1 increased from 8.37 mm^2 to 102.94 mm^2, while the crack area of the second crack increased from 15.77 mm^2 to 44.45 mm^2.

Table 27. Progression details of crack numbers 1 and 2 in concrete cube – C.

Load(kN)	**Length of 1st crack (mm)**	**Area of 1st crack (mm²)**	**Load(kN)**	**Length of 2nd crack (mm)**	**Area of 2nd crack (mm²)**
284	25.91	8.37	284	46.42	15.77
293	44.13	15.96	287	50.96	28.76
295	63.88	64.13	297	61.96	37.04
297	103.90	102.94	292	72.46	44.45

The increase in the area with respect to the increase in load for crack numbers 1 and 2 is higher as compared to the increase in the length. An increase of 12.29 times and 2.81 times is observed in the crack area of crack numbers 1 and 2 respectively. On the other hand, an increase of 4.01 times and 1.56 times in crack length is observed for crack numbers 1 and 2 respectively. The increments in the length and area of crack number are much higher than the increments observed for crack number 2. The progression of crack numbers 3 and 4 with respect to applied load in terms of crack length and area is shown in Table 28. An increase of 1.18 times is observed in the length of crack number 3 whereas a 1.87 times increase is observed in the area of the crack. Similarly, an increase of 1.33 times is observed in crack length and 2.46 times increase in crack area is observed for crack number 4.

Table 28. Progression details of crack numbers 3 and 4 in concrete cube – C.

Load(kN)	**Length of 3rd crack (mm)**	**Area of 3rd crack (mm²)**	**Load(kN)**	**Length of 4th crack (mm)**	**Area of 4th crack (mm²)**
289	55.06	15.36	290	23.94	7.85
297	58.52	20.70	298	27.65	12.62
288	62.02	25.31	270	29.40	15.76
267	65.45	28.87	263	32.00	19.38

Table 29 consists of the progression data of crack numbers 5 and 6 which are detected after the peak load of the cube. This is observed by decreasing values of load with an increase in the length and area of cracks. An increase of 1.93 times and 4.32 times are observed in crack length and area of crack number 5 respectively.

Table 29. Progression details of crack numbers 5 and 6 in concrete cube – C.

Load(kN)	**Length of 5th crack (mm)**	**Area of 5th crack (mm^2)**	**Load(kN)**	**Length of 6th crack (mm)**	**Area of 6th crack (mm^2)**
297	24.32	9.04	296	22.29	8.32
283	32.00	19.44	265	33.80	13.59
265	39.22	30.45	227	44.53	24.92
258	47.09	39.13	195	54.26	36.43

On the other hand, an increase of 2.43 times and 4.37 times are obtained in crack length and crack area of crack number 6 respectively. The increase in crack areas of crack number 5 and 6 is comparatively higher than the increments observed in the crack length of both cracks. Table 30 shows the crack length and crack area measurement with the applied compressive load for crack number 7 and 8. An increase of 2.36 times and 2.11 times are observed in crack lengths of crack numbers 7 and 8 respectively. On the other hand, an increase of 3.93 times and 3.18 times are obtained in the area of crack numbers 7 and 8 respectively.

Table 30. Progression details of crack numbers 7 and 8 in concrete cube – C.

Load(kN)	**Length of 7th crack (mm)**	**Area of 7th crack (mm^2)**	**Load(kN)**	**Length of 8th crack (mm)**	**Area of 8th crack (mm^2)**
291	41.56	14.73	288	29.18	8.83
297	61.91	30.43	276	36.96	11.43
265	81.26	42.75	255	48.93	20.22
232	98.25	57.92	241	61.77	28.08

Table 31 consists of the details of the progression in length and area of crack number 9 that occurred in the concrete cube – C. The length of crack number 9 has increased from 62.37 mm to 113.81 mm showing an increment of 1.82 times. The area of crack has increased 5.08 times. The highest increment in the crack length is obtained from crack number 1 while the largest increase in area is observed for crack number 9.

Table 31. Progression details of crack number 9 in concrete cube – C.

Load (kN)	**Length of 9th crack (mm)**	**Area of 9th crack (mm^2)**
270	62.37	26.33
225	77.65	52.20
190	100.96	100.51
155	113.81	133.92

Table 32 depicts the quantitative load and crack monitoring data of concrete cube - C. The table load, end length, end area of each crack, cumulative length and area. Considering the crack length as 150 mm, the cumulative length of all 9 cracks is 649 mm which is around 4.32 times the length of a concrete cube. Similarly, the cumulative area of all the cracks is 491.12 mm^2 which is around 2.18% of the total area of the concrete cube surface which is considered 22,500 mm^2.

Table 32. Load versus crack monitoring data for concrete cube-C.

Time (sec)	**Load (kN)**	**Crack number**	**Length of crack at the end (mm)**	**Area of crack at the end (mm^2)**	**Cumulative length of cracks (mm)**	**Cumulative area of cracks (mm^2)**
35	280.00	0	0.00	0.00	0.00	0
66	297.00	1	103.90	102.94	103.90	102.94
73	292.00	2	72.46	44.45	176.36	147.39
80	267.00	3	65.45	28.87	241.81	176.26
75	263.00	4	32.00	19.38	273.81	195.63
88	258.00	5	47.09	39.13	320.91	234.77
92	195.00	6	54.26	36.43	375.17	271.20
100	232.00	7	98.25	57.92	473.42	329.12
102	241.00	8	61.77	28.08	535.19	357.20
110	155.00	9	113.81	133.92	649.00	491.12

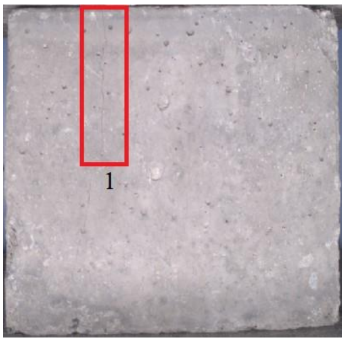
5.4.4 Concrete cube – D

Table 33 shows the crack-load analysis of the studied concrete cube - D. The data of crack number, initial or final step of cracking, load, time, crack length, crack area, location and remarks for all the five cracks which occurred on the monitored surface of the cube during the compression test. The obtained results for the cube sample show that no crack occurs till 322 kN compressive load and 50 seconds from the start of the test. The first crack initiates at 55 kN load and shows a crack length of 62.53 mm and 20.12 mm^2 crack area, while it ended at 1053 kN at 74 seconds time with a crack length of 103.57 mm and a crack area of 70.90 mm^2. The crack started from the top and continues to expand till the load of 1053 kN. During the progression, the length and area of the crack increase towards a vertically downward direction.

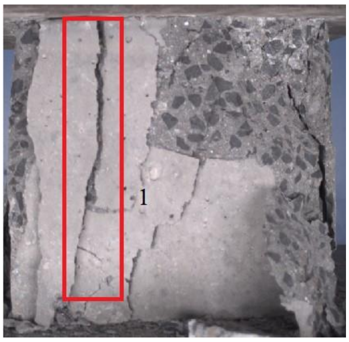
Table 33. Crack load analysis of a concrete cube-D.

Crack Number (Color)	**Steps of Cracking**	**Load (kN)**	**Time (s)**	**Crack Length (mm)**	**Crack Area (mm^2)**	**Location**
0	---	322	50	---	---	---
1 (Red)	Initial	326	55	62.53	20.12	Left
	Final	1053	74	103.57	70.90	
Remark: Visually thin vertical crack which initiates from the top edge and propagates to the bottom along with an increase in length, width and area						
2 (Blue)	Initial	919	63	75.86	25.96	Left
	Final	1090	81	97.29	34.96	
Remark: Visually thin vertical crack which initiates from top edge and propagates to the bottom along with an increase in length, width and area.						
3 (Green)	Initial	987	69	62.88	16.11	Bottom right
	Final	1307	90	169.53	197.38	
Remark: Visually thin crack at the time of initiation, which leads to spalling of the concrete crack.						
4 (Violet)	Initial	1053	75	71.83	23.20	Bottom left
	Final	1351	103	126.93	178.93	
Remark: Visible vertical crack at the time of initiation, whose length and area increase with progressive load.						
5 (Black)	Initial	1351	80	75.00	27.43	Centre
	Final	1056	108	93.43	66.51	
Remarks: Visible vertical crack at the time of initiation, whose length and area increase with progressive load.						

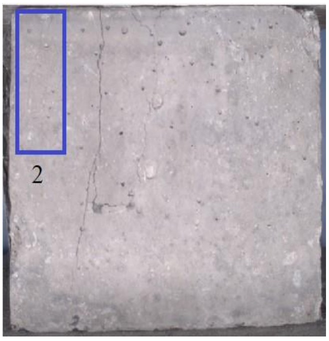
The spalling of concrete is also observed in the images of each crack shown in Figure 74. Crack number 1,2 and 3 occurred before the peak load, while crack number 4 occurs during the peak load and crack number 5 occurs after the peak load of the concrete cube. All the cracks are visually thin at the start, grow in length and area with load and propagate in a vertical downward direction. Figure 74 (a) - (i) shows the total 10 images of the concrete cube-D for the five cracks. Careful observation of these images provides the information of location, width and growth direction of all the cracks. Crack numbers 1,2, and 5 are very thin at the start which is difficult to notice with the naked eye. Spalling of concrete is observed after the initiation of crack number 1 and before starting of crack number 2.



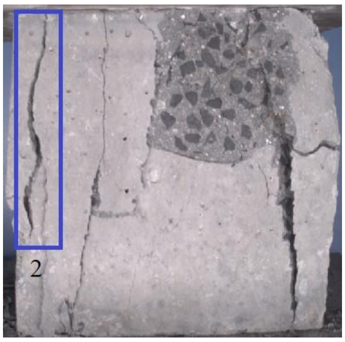
(a) Initiation of 1st crack



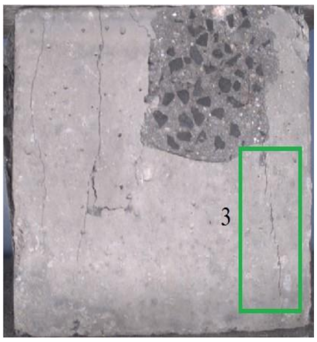
(b) End of 1st crack



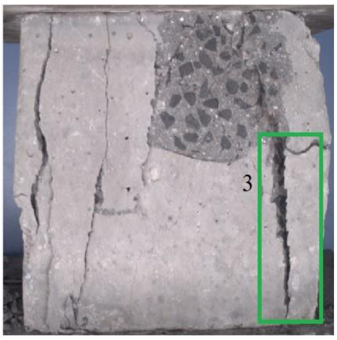
(c) Initiation of 2nd crack



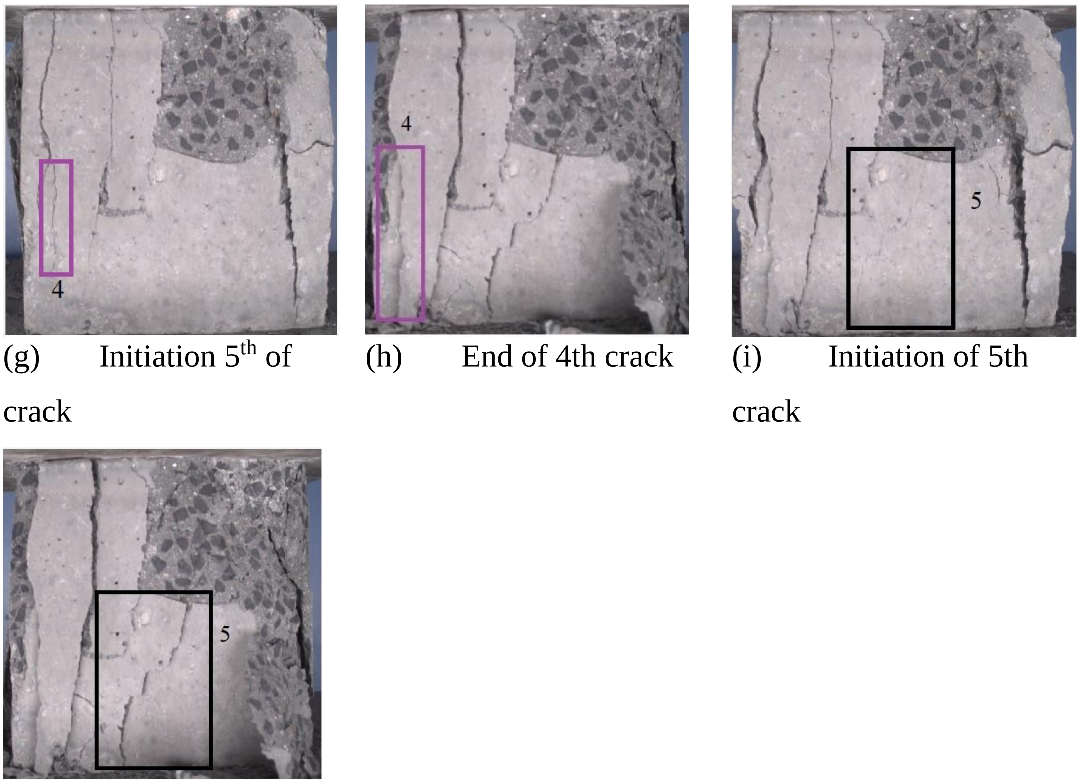
(d) End of 2nd crack



(e) Initiation of 3rd crack



(f) End of 4th crack



(g) Initiation 5th of crack

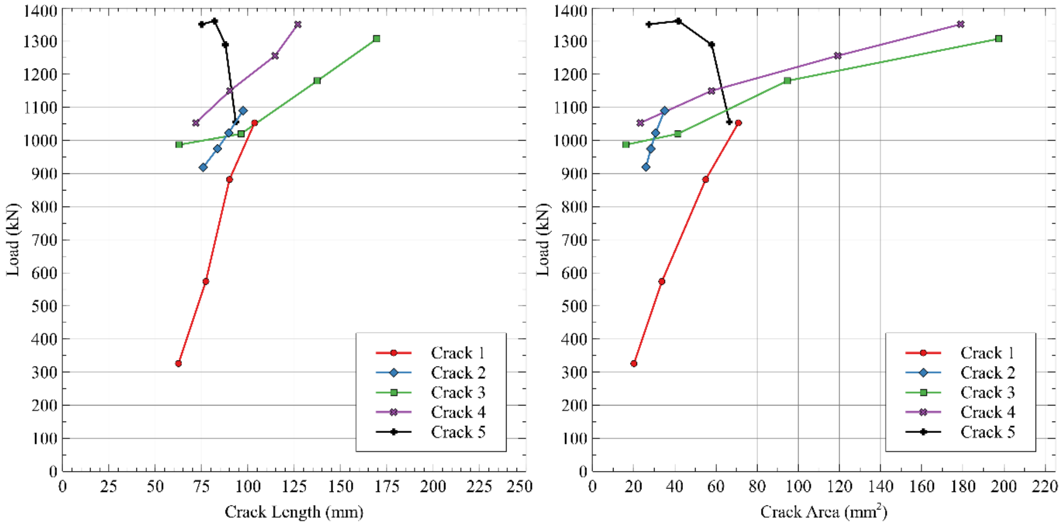
(h) End of 4th crack

(i) Initiation of 5th crack

(j) End of 5th crack

Figure 74. Concrete cube-D images showing initial crack and final crack.

The progression of individual cracks in terms of crack length and crack area plotted versus the applied load for concrete cube-D is shown in Figure 75 (a) and Figure 75 (b) respectively. Four data points of each crack are shown to understand the progression of individual cracks. Crack number 1,2,3 and 4 shows an increasing trend for length and area with respect to load. However, crack number 5 shows a downward trend because the load is decreasing with an increase in length and area. The slope of increment observed in the length and area of each of the cracks is different.



(a) Individual crack length versus load (b) Individual crack area versus load

Figure 75. Progressive and individual crack monitoring with applied load for cube-D.

Table 34, Table 35, and Table 36 depict the progression of each crack with reference to load, crack length and crack area. An increase of 1.65 times and 1.28 times in crack length is observed for crack numbers 1 and 2 respectively. For crack area, an increase of 3.52 times and 1.34 times is observed in crack numbers 1 and 2 respectively.

Table 34. Progression details of crack numbers 1 and 2 in concrete cube - D.

Load(kN)	**Length of 1st crack (mm)**	**Area of 1st crack (mm²)**	**Load(kN)**	**Length of 2nd crack (mm)**	**Area of 2nd crack (mm²)**
326	62.53	20.12	919	75.86	25.96
574	77.25	33.71	975	83.53	28.32
882	90.03	54.99	1023	89.63	30.67
1053	103.57	70.90	1090	97.29	34.96

The progression of crack numbers 3 and 4 with respect to applied load in terms of crack length and area is shown in Table 35. A large increase in the area of crack numbers 3 and 4 is observed during the progression with respect to load. The area of crack number 3 starts at 16.11 mm^2 and reaches 197.38 mm^2 which is an increment of 12.25 times. The area of the 4th crack also shows an increment of 7.71 times with respect to the initial area. An increment of 2.69 and 1.76 times is observed for crack length of cracks 3 and 4 respectively. The increment in the crack area of both cracks is comparatively higher than the length of both cracks.

Table 35. Progression details of crack numbers 3 and 4 in concrete cube – D.

Load(kN)	**Length of 3rd crack (mm)**	**Area of 3rd crack (mm^2)**	**Load(kN)**	**Length of 4th crack (mm)**	**Area of 4th crack (mm^2)**
987	62.88	16.11	1053	71.83	23.20
1020	96.26	41.45	1150	90.26	57.80
1180	137.38	94.59	1256	114.63	119.22
1307	169.53	197.38	1351	126.93	178.93

Table 36 consists of the progression data of crack number 5 of the cube - D. The length of crack 5 has increased 1.24 times while the area has shown an increment of 2.42 times. Out of all the five cracks, crack number 3 has shown the highest increment in length and area.

Table 36. Progression details of crack number 5 in concrete cube – D.

Load(kN)	**Length of 5th crack (mm)**	**Area of 5th crack (mm^2)**
1351	75.00	27.43
1361	81.94	41.76
1289	87.90	57.95
1056	93.43	66.51

Table 37 shows the data of time, load, length of the crack, and area of the crack at the end of the crack along with the cumulative data of length and area measured for the crack which occurred on the concrete cube - D.

Table 37. Load versus crack monitoring data for concrete cube-D.

Time (sec)	Load (kN)	Crack number	Length of crack at the end (mm)	Area of crack at the end (mm^2)	Cumulative length of cracks (mm)	Cumulative area of cracks (mm^2)
50	322	0	0.00	0.00	0.00	0
74	1053	1	103.57	70.90	103.57	70.90
81	1090	2	97.29	34.96	200.86	105.86
90	1307	3	169.53	197.38	370.40	303.24
103	1351	4	126.93	178.93	497.33	482.17
108	1056	5	93.43	66.51	590.76	548.68

Considering the crack length as 150 mm, the cumulative length of all 5 cracks is 590.76 mm which is around 3.93 times the length of the concrete cube considered as 150 mm. Similarly, the cumulative area of all the cracks is 548.68 mm^2 which is around 2.43% of the total area of the concrete cube surface which is considered as 22,500 mm^2. The methodology of real-time crack monitoring in synchronization with loading is implemented during compression testing of concrete cubes. However, it can be easily implemented in real-life structures like buildings and bridges. Multiple cameras can be installed near highly stressed areas of concrete structures for capturing images at regular intervals of time. These images can be communicated to the computing units for further processing through a wired or wireless network. Further, the CNN based intelligent algorithms for crack detection to identify cracks in captured images. Subsequently, various parameters of cracks like orientation, length, width, and area are obtained. The complete system of image acquisition and crack identification can provide automated inspection and monitoring of structural health.

For load measurement load cells need to be installed at the support during the construction of structures. Alternately, electrical resistant strain gauges can be installed, so that from strain measurements forces acting at critical locations can be estimated. Advanced image acquisition methods like 3D laser profiles and apparatus like unmanned aerial vehicles can also be utilized to capture images of structures which are not easily accessible to a human. With the advances in technology related to computation, communication, AI, and algorithms, cost-effective and efficient inspection of concrete structures is becoming feasible. Application of computer vision methodology will be utilized to a greater extent for structural health monitoring of structures in the upcoming future.

List of Publications

Journal Publications

1. Kapadia Harsh K, Patel Paresh V, Patel Jignesh B., “Monitoring and analysis of crack developments in concrete using machine vision”. *Journal of Structural Engineering (SERC),* 49 (3), 204-222, (2022).
2. Kapadia Harsh, Patel Paresh V, Patel Jignesh B., Kanani Nikhil, “Implementation of Computer Vision technique for Crack Monitoring in Concrete Structure”. *Journal of The Institution of Engineers (India): Series A Civil, Architectural, Environmental and Agricultural Engineering,* 104, 111-123 (2023).
3. Kapadia Harsh K, Patel Paresh V, Patel Jignesh B., “Convolutional neural network based improved crack detection in concrete cubes”. *International Journal of Computing and Digital Systems,* 13 (1), 341-352, (2023).

Conference Publications

1. Nikhil Kanani, Paresh V Patel, Harsh Kapadia. “Convolutional neural network based technique for accurate detection of cracks in concrete.” In 2021 International conference on research and development in civil engineering, organized by Department of Civil Engineering, SVKM’s Institute of Technology, Dhule, in Association with Indian Society for Technical education in Virtual Mode, 2021.
2. Pushpit Jain, Rishi Hiran, Harsh Kapadia, Paresh Patel, and Jignesh B. Patel. "Hybrid Computing System for Real-Time Implementation of a Convolutional Neural Network Application." In 2022 International Conference for Advancement in Technology (ICONAT), pp. 1-6. IEEE, 2022.
3. Harsh Kapadia, Nikhil Kanani, Paresh V Patel, Jignesh B Patel. “Accurate crack identification on the concrete structure using convolutional neural network.” In 2022 3rd International conference on new horizons in green civil engineering, organized by University of Victoria and BC Housing, Canada, 2022.

Book chapter Publications

1. Kapadia, H., Patel, R., Shah, Y., Patel, J.B., Patel, P.V. (2019). An Improved Image Pre-processing Method for Concrete Crack Detection. In: Pandian, D., Fernando, X., Baig, Z., Shi, F. (eds) Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB). ISMAC 2018. Lecture Notes in Computational Vision and Biomechanics, vol 30. Springer, Cham.
2. Kapadia, H., Soneji, N., Rotti, A., Patel, P.V., Patel, J.B. (2023). Monitoring and Analysis of Surface Cracks in Concrete Using Convolutional Neural Network. In: Marano, G.C., Rahul, A.V., Antony, J., Unno Kartha, G., Kavitha, P.E., Preethi, M. (eds) Proceedings of SECON'22. SECON 22. Lecture Notes in Civil Engineering, vol 284. Springer, Cham.

References

3 Methods for Crack Depth Measurement in Concrete. 31 May 2017. 12 July 2022. <https://www.fprimec.com/3-methods-crack-depth-measurement-in-concrete/>.

(10PCS) Muka Plastic Concrete Crack Width Ruler-Gloss. 12 July 2022. 12 July 2022. <https://www.amazon.com/10PCS-Muka-Plastic-Concrete-Ruler-Gloss/dp/B08P4PPF4F>.

A. Vedaldi, and Lenc, K. "MatConvNet: convolutional neural networks for MATLAB." *23rd ACM International Conference on Multimedia*. Brisbane, Australia, 2015. 689-692.

Ahmed Mahgoub Talab, , Zhangcan Huang, Fan Xi, and Liu HaiMing. "Detection crack in image using Otsu method and multiple filtering in image processing techniques." *Optik-International Journal for Light and Electron Optics* 127.3 (2016): 1030-1033.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *In Advances in neural information processing systems*. Nevada, USA, 2012. 1097-1105.

Ali Raza Asif, Asim Waris, Syed Omer Gilani, Mohsin Jamil, Hassan Ashraf, Muhammad Shafique, and Imran Khan Niazi. "Performance evaluation of convolutional neural network for hand gesture recognition using EMG." *Sensors* 20.6 (2020): 1642.

Allen Zhang, Kelvin CP Wang, Yue Fei, Yang Liu, Siyu Tao, Cheng Chen, Joshua Q. Li, and Baoxian Li. "Deep learning–based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet." *Journal of Computing in Civil Engineering* 32.5 (2018): 04018041.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision." n.d. *https://arxiv.org*. 26 April 2020. <http://arxiv.org/abs/1704.04861>.

Andrushia, A. Diana, et al. "Shrinkage Crack Detection in Expansive Soil using Deep Convolutional Neural Network and Transfer Learning." *KSCE Journal of Civil Engineering* 26.10 (2022): 4202-4213.

Andrushia, Diana, et al. "Autonomous detection of concrete damage under fire conditions." *Automation in Construction* 140 (2022): 104364.

Andrushia, Diana. and and G. P. Arulraj N. Anand. "A novel approach for thermal crack detection and quantification in structural concrete using ripplet transform." *Struct Control Health Monit* 11 (2020): e2621.

Arabinda Sharma, Neeraj Mehta. “Structural health monitoring using image processing techniques- A review.” *International journal of modern computer science* 4.4 (2016): 93-97.

Arun Mohan, and Sumathi Poobal. “Crack detection using image processing: A critical review and analysis.” *Alexandria Engineering Journal* (2017).

Atsushi Ito, Yoshimitsu Aoki, and Shuji Hashimoto. “Accurate extraction and measurement of fine cracks from concrete block surface image.” *28th Annual Conference of the IEEE Industrial Electronics*. IEEE, 2002. 2202-2207.

Ayaha Miyamoto, Masa-Aki Konno, Eugen Bruhwiler. “Automatic crack recognition system for concrete structures using image processing approach.” 6.5 (2007).

Bang Yeon Lee, , Yun Yong Kim, Seong-Tae Yi, and Jin-Keun Kim. “Automated image processing technique for detecting and analyzing concrete surface cracks.” *Structure and Infrastructure engineering* 9.6 (2013): 567-577.

Barzin Mobasher, Michael S. Mamlouk, and How-Ming Lin. “Evaluation of Crack Propagation Properties of Asphalt Mixtures.” *Journal of Transportation Engineering* 123.5 (1997): 405-413.

Basler Cable USB 3.0, Micro B sl/A, P, 3 m - Data Cable. n.d. Basler. 21 October 2022. <https://www.baslerweb.com/en/products/cable/basler-cable-usb-3-0-micro-b-sl-a-p-3-m/>.

Besnard, Gilles, François Hild, and Stéphane Roux. ““Finite-element” displacement fields analysis from digital images: application to Portevin–Le Châtelier bands.” *Experimental mechanics* 46 (2006): 789-803.

Billie F Spencer Jr, Vedhus Hoskere, Yasutaka Narazaki. “Advances in computer vision-based civil infrastructure inspection and monitoring.” *Engineering* 5.2 (2019): 199-222.

Cao Meili, Qiu Xingjun. “"Pixel Measurement for Cracks in Concrete Construction.” *2013 International Conference on Computational and Information Sciences*. IEEE, 2013. 1920-1923.

Carlo Tomasi, and Roberto Manduchi. “Bilateral filtering for gray and color images.” *Sixth international conference on computer vision*. IEEE, 1998. 839-846.

Chaitanya Nagpal, Shiv Ram Dubey. “A performance evaluation of convolutional neural networks for face anti spoofing.” *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019. 1-8.

Changyon Kim, Jeonghwan Gwak, Daeyoung Shim, and Moongu Jeon. “A framework for automatically constructing a dataset for training a vehicle detector.” *International Journal of Computational Vision and Robotics* 9.2 (2019): 192-206.

REFERENCES

Chaobo Zhang, Maziar Jamshidi, Chih-Chen Chang, Xiaojun Liang, Zhiwen Chen, and Weihua Gui. “Concrete Crack Quantification using Voxel-Based Reconstruction and Bayesian Data Fusion.” *EEE Transactions on Industrial Informatics* 18.11 (2022): 7512-7524.

Chen Feng, Ming-Yu Liu, Chieh-Chi Kao, and Teng-Yok Lee. “Deep Active Learning for Civil Infrastructure Defect Detection and Classification.” *Computing in civil engineering* (2017): 298-306.

Chia-Chi Cheng, Tao-Ming Cheng, Chih-Hung Chiang. “Defect detection of concrete structures using both infrared thermography and elastic waves.” *Automation in concrete* (2008): 87-92.

Christian Koch, Kristina Georgieva, Varun Kasireddy, Burcu Akinci, and Paul Fieguth. “A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure.” *Advanced Engineering Informatics* 29.2 (2015): 196-210.

Christin Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. “Rethinking the inception architecture for computer vision.” *Proceedings of the IEEE conference on computer vision and pattern recognition*. Nevada, USA: IEEE, 2016. 2818-2826.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. “Going deeper with convolutions.” *IEEE conference on computer vision and pattern recognition*. IEEE, 2015. 1-9.

Chunyan Shao, Limin Zhang, and Wang Pan. “PTZ Camera-Based Image Processing for Automatic Crack Size Measurement in Expressways’.” *IEEE Sensors Journal* 21.20 (2021): 23352-23361.

Cipolla Roberto, Alex Pentland. *Computer vision for human-machine interaction*. Cambridge university press, 1998.

computer-vision. 09 07 2022. IBM. 09 july 2022. <https://www.ibm.com/in-en/topics/computer-vision>.

Concrete Cracking: Evaluating width, depth and movement. Ed. Ben Backus. 12 July 2022. 12 July 2022. <https://www.globalgilson.com/blog/concrete-cracking>.

Construction Materials Testing Equipment. 2016. humboldtmfg. 16 December 2016. <https://www.humboldtmfg.com/concrete-crack-monitor-crack-gauge.html>.

daA2500-14uc (CS-Mount) - Basler dart. n.d. Basler. 21 October 2022. <https://www.baslerweb.com/en/products/cameras/area-scan-cameras/dart/daa2500-14uc-cs-mount/>.

Danfeng Xie, Lei Zhang, and Li Bai. “Deep learning in visual computing and signal processing.” *Applied Computational Intelligence and Soft Computing* (2017).

David White, James D. Dunn, Alexandra C. Schmid, Richard I. Kemp. “Error rates in users of automatic face recognition software.” *PLoS ONE* 10.10 (2015).

Dhanya s pankaj, Rama rao nidamanuri, R bhanu prasad. “3-D imaging techniques and review of products.” International conference on innovation in computer science and engineering ICICSE 2013, 2013.

Diana Andrushia, A., and N. Anand. “Auto-associative neural network based concrete crack detection.” *Smart Computing Paradigms: New Progresses and Challenges*. Springer, 2020. 239-244.

Diana, Andrushia A., et al. “Automatic Detection of Surface Thermal Cracks in Structural Concrete with Numerical Correlation Analysis.” Naser, M. Z. *Leveraging Artificial Intelligence in Engineering, Management, and Safety of Infrastructure*. CRC Press, 2022. 120-139.

Diana, Andrushia and Eva Lubloy. “Deep learning based thermal crack detection on structural concrete exposed to elevated temperature.” *Advances in Structural Engineering* 24.9 (2021): 1896-1909.

Diana, Andrushia, N. Anand and Prince Arulraj. “Anisotropic diffusion based denoising on concrete images and surface crack segmentation.” *International Journal of Structural Integrity* (2019).

Different types of concrete. 28 March 2020. 12 April 2023. <https://www.dcpu1.com/blog/different-types-of-concrete/>.

Dihao Ai, Guiyuan Jiang, Siew-Kei Lam, Peilan He, and Chengwu Li. “Computer vision framework for crack detection of civil infrastructure—A review.” *Engineering Applications of Artificial Intelligence* 117 (2023): 105478.

Dong, Chuan-Zhi, and F. Necati Catbas. “A review of computer vision–based structural health monitoring at local and global levels.” *Structural Health Monitoring* 20.2 (2021): 692-743.

Donghan Lee, , Jeongho Kim, and Daewoo Lee. “Robust Concrete Crack Detection Using Deep Learning-Based Semantic Segmentation.” *International Journal of Aeronautical and Space Sciences* 20.1 (2019): 287-299.

Dorafshan, Sattar and and Marc Maguire Robert J. Thomas. “Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete.” *Construction and Building Materials* 186 (2018): 1031-1045.

REFERENCES

Dulari Bhatt, Chirag Patel, Hardik Talsania, Jigar Patel, Rasmika Vaghela, Sharnil Pandya, Kirit Modi, and Hemant Ghayvat. “CNN variants for computer vision: History, architecture, application, challenges and future scope.” *Electronics* 10.20 (2021): 2470.

Eberechi Ichi, Faezeh Jafari, and Sattar Dorafshan. “SDNET2021: Annotated NDE Dataset for Subsurface Structural Defects Detection in Concrete Bridge Decks.” *Infrastructures* 7.9 (2022): 107.

Edge TPU performance benchmarks. n.d. Coral (Google). 22 October 2022. <https://coral.ai/docs/edgetpu/benchmarks/>.

elcometer-143-crack-width-ruler. n.d. 07 July 2022. <https://www.elcometer.com/en/concrete-inspection/elcometer-143-crack-width-ruler.html>.

Eleni N Chatzi, Badri Hiriyur, Haim Waisman, and Andrew W. Smyth. “Experimental application and enhancement of the XFEM–GA algorithm for the detection of flaws in structures.” *Computers & Structures* 89.7-8 (2011): 556-570.

Elgendy, Mohamed. *Deep learning for vision systems*. Simon and Schuster, 2020.

Fu-Chen Chen, and Mohammad R. Jahanshahi. “NB-CNN: deep learning-based crack detection using convolutional neural network and naive Bayes data fusion.” *IEEE Transactions on Industrial Electronics* 65.5 (2018): 4392-4400.

Gajanan K Choudhary, Sayan Dey. “Crack Detection in Concrete Surfaces using Image Processing, Fuzzy logic and Neural network.” *IEEE fifth International Conference on Advanced Computational Intelligence(ICACI)*. Nanjing, Jiangsu, China, 2012.

Gopalakrishnan, Kasthurirangan, Siddhartha K. Khaitan, Alok Choudhary, and Ankit Agrawal. “Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection.” *Construction and building materials* 157 (2017): 322-330.

Gurmu, Argaw Tarekegn, Adam Krezel, and Citra Ongkowijoyo. “Fuzzy-stochastic model to assess defects in low-rise residential buildings.” *Journal of Building Engineering* 40 (2021): 102318.

Hafiz Sulaiman Munawar, Hammad, Ahmed WA Assed Haddad, Carlos Alberto Pereira Soares, S. Travis Waller. “Image-Based Crack Detection Methods: A Review.” *Infrastructures* 6.8 (2021): 115.

Han Hu, Quanquan Gu, Jie Zhou. “HTF: a novel feature for general crack detection.” *2010 IEEE International Conference on Image Processing*. IEEE, 2010. 1633-1636.

Henrique Oliveira, Paulo Lobato Correia. “Automatic Road Crack Detection and Characterization.” *IEEE Transactions on Intelligent Transportation Systems* 14.1 (2012): 155-168.

Hongk Xu, Yuan Tian, Shan Lin and Shichao Wang. “Research of Image Segmentation Algorithm Applied to Concrete Bridge Cracks.” *International Conference on Information Science and Technology (ICIST)*. IEEE, 2013. 1637-1640.

Hongyan Xu, Su Xiu, Yi Wang, Huaiyu Cai, Kerang Cui, and Xiaodong Chen. “Automatic Bridge Crack Detection Using a Convolutional Neural Network.” *Applied Sciences* 9.14 (2019): 2867.

Hongyu Zhu, Mohamed Akrout, Bojian Zheng, Andrew Pelegris, Anand Jayarajan, Amar Phanishayee, Bianca Schroeder, and Gennady Pekhimenko. “Benchmarking and analyzing deep neural network training.” *IEEE International Symposium on Workload Characterization (IISWC)*. Raleigh, North Carolina, USA: IEEE, 2018. 88-100.

Huang, Pingrang Wang and Hongwei. “Comparison Analysis on Present Image-based Crack Detection methods in Concrete Structure.” *3rd International Congress on Image and Signal Processing (CISP)*. IEEE, 2010. 2530-2533.

Hung, Po-Chih, and A. S. Voloshin. “In-plane strain measurement by digital image correlation.” *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 25 (2003): 215-221.

Hussain Rasheed, Sherali Zeadally. “Autonomous cars: Research results, issues, and future challenges.” *IEEE Communications Surveys & Tutorials* 21.2 (2019): 1275-1313.

Hyo Seon Park, Hwan Young Lee, Se Woon Choi, and Yousok Kim. “A practical monitoring system for the structural safety of mega-trusses using wireless vibrating wire strain gauges.” *Sensors* 13.12 (2013): 17346 - 17361.

Industrial USB to RS485 Bidirectional Converter. n.d. Waveshare Electronics. 21 October 2022. <https://www.waveshare.com/usb-to-rs485.htm>.

Intel® Neural Compute Stick 2 (Intel® NCS2). n.d. Intel. 22 October 2022. <https://www.intel.com/content/www/us/en/developer/tools/neural-compute-stick/overview.html>.

Ito Atsushi, Yoshimitsu Aoki, and Shuji Hashimoto. “Accurate extraction and measurement of fine cracks from concrete block surface image.” *Proceedings of the 2002 28th Annual Conference of the IEEE Industrial Electronics*. 2002. 2202-2207.

REFERENCES

JaChing Chou, Wende A. O'Neill, and H. D. Cheng. “Pavement distress classification using neural networks.” *In Proceedings of IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 1994. 397-401.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database.” *IEEE conference on computer vision and pattern recognition,*. Miami, FL, USA: IEEE, 2009. 248-255.

Jianghua Deng, Amardeep Singh, Yiyi Zhou, Ye Lu, and Vincent Cheng-Siong Lee. “Review on computer vision-based crack detection and quantification methodologies for civil structures.” *Construction and Building Materials* 356 (2022): 129238.

Jinlong Chen, Xiaochuan Zhang, Nan Zhan, and Xiaoyan Hu. “Deformation measurement across crack using two-step extended digital image correlation method.” *Optics and Lasers in Engineering* 48.11 (2010): 1126-1131.

Jung Jin Kim, Ah-Ram Kim, and Seong-Won Lee. “Artificial neural network-based automated crack detection and analysis for the inspection of concrete structures.” *Applied Sciences* 10.22 (2020): 8105.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition.” *IEEE conference on computer vision and pattern recognition*. Nevada, USA: IEEE, 2016. 770-778.

Karen Simonyan, and Andrew Zisserma. “Very deep convolutional networks for large-scale image recognition.” *International Conference on Learning Representations 2014*. San Diego, USA, 2014. 1409-1556.

Khalili, K., and M. Vahidnia. “Improving the accuracy of crack length measurement using machine vision.” *Procedia Technology* 19 (2015): 48-55.

k-NN classifier for image classification. 08 August 2016. 11 May 2017. <http://www.pyimagesearch.com/2016/08/08/k-nn-classifier-for-image-classification/>.

Kumar, Babloo, and Sayantari Ghosh. “Detection of Concrete Cracks Using Dual channel Deep Convolutional Network.” *11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2020. 1-7.

Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. “Review of deep learning:

Concepts, CNN architectures, challenges, applications, future directions." *Journal of big Data* 8.1 (2021): 1-74.

Lalit Prakash Saxena. "Niblack's binarization method and its modifications to real-time applications: a review." *Artificial Intelligence Review* 51.4 (2019): 673-705.

"Laser triangulation." n.d. *LMI Technologies.* 20 March 2017. <https://lmi3d.com/laser-triangulation>.

Liang-Chien, Chen, Yi-Chen Shao, Huang-Hsiang Jan, Chen-Wei Huang, Yong-Ming Tien. "Measuring system for cracks in concrete using multitemporal images." *Journal of Surveying Engineering* 132.2 (2006): 77-82.

load indicator. n.d. Rudrra Sensors. 21 October 2022. <https://www.rudrra.com/products/load-indicator.html>.

Low profile compression load cell (RTN). n.d. Rudrra Sensors. 21 October 2022. <https://www.rudrra.com/products/load-cell/compression-load-cell/low-profile-compression-load-cell.html>.

M. J. Anitha, R. Hemalatha, and S. Radha. "A Survey on Crack Detection Algorithms for Concrete Structures." *Advances in Smart System Technologies* (2021): 639-654.

M. M. Sarker, T. A. Ali, A. Abdelfatah, S. Yehia, and A. Elaksher. "A cost-effective method for crack detection and measurement on concrete surface." *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 42 (2017): 237.

Mahbub Hussain, Jordan J. Bird, and Diego R. Faria. "A study on CNN transfer learning for image classification." *In UK Workshop on computational Intelligence*. UK: Springer, Cham, 2018. 191-202.

Martin Mundt, Sagnik Majumder, Sreenivas Murali, Panagiotis Panetsos, and Visvanathan Ramesh. "Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2019. 11196-11205.

Matthias Breier, Philipp Möller, Wei Li. "Accurate laser triangulation using a perpendicular camera setup to assess the height profile of PCBs." Seville, Spain: 2015 IEEE International Conference on Industrial Technology (ICIT), 2015.

Md Zawad, Rahat Shahriar, Md Zawad, Fahad Shahriar, Md Rahman, and Sudipto Nath Priyom. "A comparative review of image processing based crack detection techniques on civil engineering structures." *Journal of Soft Computing in Civil Engineering* 5.3 (2021): 58-74.

meta-learning-CODEBRIM. n.d. Github. 12 April 2022. <https://github.com/ccc-frankfurt/meta-learning-CODEBRIM>.

Milind G, Padalkar, Manjunath V. Joshi and Nilay L. Khatri. "Auto-inpainting Cracks in Heritage Scenes." Milind G. Padalkar, Manjunath V. Joshi & Nilay L. Khatri. *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography and Imaging book series (SLVCCGACPI)*. Springer, 2017. 93-129.

Milind G, Padalkar, Mukesh A. Zaveri and Manjunath V., Joshi. "Svd based automatic detection of target regions for image inpainting." *Asian Conference on Computer Vision*. Berlin, Heidelberg: Publisher, 2012. 61-71.

Mohamad Alipour, , Devin K. Harris, and Gregory R. Miller. "Robust Pixel-Level Crack Detection Using Deep Fully Convolutional Neural Networks." *Journal of Computing in Civil Engineering* 33.6 (2019): 04019040.

MPZ Machine Vision Series H0514-MP2. n.d. Computar . 21 October 2022. <https://computar.com/product/551/H0514-MP2>.

N. A. M. Yusof, A. Ibrahim, M. H. M. Noor, N. M. Tahir, N. M. Yusof, N. Z. Abidin, and M. K. Osman. "Deep convolution neural network for crack detection on asphalt pavement." *Journal of Physics: Conference Series* 1349.1 (2019): 012020.

Narges Kheradmandi, and Vida Mehranfar. "A critical review and comparative study on image segmentation-based techniques for pavement crack detection." *Construction and Building Materials* 321 (2022): 126162.

Nicola Gehri, Jaime Mata-Falcón, and Walter Kaufmann. "Refined extraction of crack characteristics in Large-scale concrete experiments based on digital image correlation." *Engineering Structures* 251 (2022): 113486.

Nobuyuki Otsu. "A threshold selection method from gray-level histograms." *EEE transactions on systems, man, and cybernetics* 9.1 (1979): 62-66.

Nur Sila Gulgec, Martin Takáč, and Shamim N. Pakzad. "Convolutional Neural Network Approach for Robust Structural Damage Detection and Localization." *Journal of Computing in Civil Engineering* 33.3 (2019).

Özgenel, Ç. F., and A. Gönenç Sorguç. “Performance comparison of pretrained convolutional neural networks on crack detection in buildings.” *Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC)*. Berlin, Germany: IAARC Publications, 2018. 1-8.

Padalkar, Milind G and Manjunath V. Joshi. “Auto-inpainting heritage scenes: a complete framework for detecting and infilling cracks in images and videos with quantitative assessment.” *Machine Vision and Applications* 26.2-3 (2015): 317-337.

Padalkar, Milind G. and and Alessio Del Bue Carlos Beltrán-González. “Multi-illumination fusion with crack enhancement using cycle-consistent losses.” *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021. 2898-2902.

Paranjape, Raman B. “Fundamental Enhancement Techniques.” *In Biomedical Engineering, Handbook of Medical Imaging*. Academic Press, 2000. 3-18.

Pauly, Leo, D. Hogg, R. Fuentes, and H. Peel. “Deeper Networks for Pavement Crack Detection.” *Proceedings of the 34th International Symposium in Automation and Robotics in Construction ISARC*. Taipei Taiwan, 2017. 479-485.

Philipp Hüthwohl, Ruodan Lu, and Ioannis Brilakis. “Multi-classifier for reinforced concrete bridge defects.” *Automation in Construction* 105 (2019): 102824.

Pingrang Wang, Hognwei Huang. “Comparison analysis on present image based crack detection methods in concrete structures.” *3rd International congress on Image and Signal processing ICISP2010*. China, 2010.

Qipei Mei, and Mustafa Gül. “A cost effective solution for pavement crack inspection using cameras and deep neural networks.” *Construction and Building Materials* 256 (2020): 119397.

R. S. Adhikari, O. Moselhi, and A. Bagchi. “Image-based retrieval of concrete crack properties for bridge inspection.” *Automation in construction* 39 (2014): 180-194.

Raj, P. Purushothama. *Building construction materials and techniques*. Pearson Education India, 2017.

Rameshwar ashok wagh, Dr. M S Panse, Hemant Apte. “Calibration methods for height measurement of object using laser triangulation.” 3.4 (2015).

S. Xua, J. Wang, X. Wang, W. Shou. “Computer vision techniques in construction, operation and maintenance phases of civil assets: A critical review.” *36th International Symposium on Automation and Robotics in Construction (ISARC 2019)*. 2019.

REFERENCES

Safiuddin, Md, ABM Amrul Kaish, Chin-Ong Woon, and Sudharshan N. Raman. “Early-age cracking in concrete: Causes, consequences, remedial measures, and recommendations.” *Applied Sciences* 8.10 (2018): 1730.

Sankar K Pal, Ashish Gosh, Malay K Kundu. “Soft computing and Image analysis:Features, relevance and hybridization.” (2000).

Sattar Dorafshan, Robert J. Thomas, and Marc Maguire. “SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks.” *Data in brief* 21 (2018): 1664-1668.

Seung-Nam Yu, Jae-Ho Jang, and Chang-Soo Han. “Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel.” *Automation in Construction* 16.3 (2007): 255-261.

Shahid Kabir, Patrice Rivard. “Damage classification of concrete structures based on grey level co-occurrence matrix using Haar\'s discrete wavelet transform.” *Computers and Concrete* 4.3 (2007): 243-257.

Shane Barratt, Rishi Sharma. “A note on the inception score.” *arXiv preprint arXiv* (2018): 1801.01973.

Shankar H. Sanni, Keshavraj. Girinivas. “Analytical investigation on reinforced concrete beams.” *International Research Journal of Engineering and Technology* 5.12 (2018): 1509-1513.

Shengyuan Li, Zhao Xuefeng. “Image-Based Concrete Crack Detection Using Convolutional Neural Network and Exhaustive Search Technique.” *Advances in Civil Engineering* (2019).

Shuai Zhao, Dongming Zhang, Yadong Xue, Mingliang Zhou, and Hongwei Huang. “A deep learning-based approach for refined crack evaluation from shield tunnel lining images.” *Automation in Construction* 132 (2021): 103934.

Somin Park, Seongdeok Bang, Hongjo Kim, and Hyoungkwan Kim. “Patch-Based Crack Detection in Black Box Images Using Convolutional Neural Networks.” *Journal of Computing in Civil Engineering* 33.3 (2019): 04019017.

Steger Carsten, Markus Ulrich, Christian Wiedemann. *Machine vision algorithms and applications*. John Wiley & Sons, 2018.

Stephanie German, Ioannis Brilakis, and Reginald DesRoches. “Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments.” *Advanced Engineering Informatics* 26.4 (2012): 846-858.

Structural Defects Network (SDNET) 2018. n.d. 22 October 2022. <https://www.kaggle.com/datasets/aniruddhsharma/structural-defects-network-concrete-crack-images>.

Suguru Yokoyama, and Takashi Matsumoto. “Development of an automatic detector of cracks in concrete using machine learning.” *International conference on Sustainable Civil Engineering Structures and Construction Materials, SCESCM, 2016*. Bali, Indonesia: Procedia engineering, Elsevier, 2017. 1250-1255.

Sun, Yaofeng, John HL Pang, Chee Khuen Wong, and Fei Su. “Finite element formulation for a digital image correlation method.” *Applied optics* 44.34 (2005): 7357-7363.

Suresh Bhalla, Tuli, S Arora, R. “Defect detection in concrete structures using thermal imaging techniques.” *Experimental Techniques* 35 (2011): 39-43.

Suriani Mohd Sam, Kamilia Kamardin, Nilam Nur Amir Sjarif, and Norliza Mohamed. “Offline signature verification using deep learning convolutional neural network (CNN) architectures GoogLeNet Inception-v1 and Inception-v3.” *Procedia Computer Science* 161 (2019): 475-483.

Sylvie Chambon, Peggy Subirats, and Jean Dumoulin. “. Introduction of a Wavelet Transform Based on 2D Matched Filter in a Markov Random Field for Fine Structure Extraction: Application on Road Crack Detection.” *In Image Processing: Machine Vision Applications II SPIE* 7251 (2009): 87-98.

Syska Slim LED 15W RDL Square Downlight-6500K. n.d. Syska. 21 October 2022. <https://syska.co.in/syska-slim-led-15w-rdl-square-downlight-6500k.html>.

T. Barkavi, Natarajan Chidambarathanu. “Processing Digital Image for Measurement of Crack Dimensions in Concrete.” *Civil Engineering Infrastructures Journal* 52.1 (2019): 11-22.

Talaat, A., Emad, A., Tarek, A., Masbouba, M., Essam, A., & Kohail, M. “Factors affecting the results of concrete compression testing: A review.” *Ain Shams Engineering Journal* 12.1 (2021): 205-221.

TensorFlow models on the Edge TPU. n.d. Coral (Google). 20 October 2022. <https://www.coral.ai/docs/edgetpu/models-intro>.

Tomoyuki Yamaguchi, Shingo Nakamura, Ryo Saegusa, and Shuji Hashimoto. “Image based crack detection for real concrete surfaces.” *IEEJ Transactions on Electrical and Electronic Engineering* 3.1 (2008): 128-135.

REFERENCES

Tran Hiep Dinh, Q. P. Ha, and Hung M. La. “Computer vision-based method for concrete crack detection.” *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. Phuket, Thailand: IEEE, 2016. 1-6.

Tsang, Sik-Ho. *Review: Inception-v3 — 1st Runner Up (Image Classification) in ILSVRC 2015*. 10 September 2018. 02 July 2019. <https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c>.

Tung-Ching Su, Ming-Der Yang. “Morphological segmentation based on edge detection-II for automatic concrete crack measurement.” *Computers and Concrete* 21.6 (2018): 727-739.

Tuong Lam Nguyen, Stephen A. Hall, Pierre Vacher, and Gioacchino Viggian. “Fracture mechanisms in soft rock: identification and quantification of evolving displacement discontinuities by extended digital image correlation.” *Tectonophysics* 503.1-2 (2011): 117-128.

Understanding laser based 3-D triangulation methods. 5 June 2015. 28 February 2017. <http://www.vision-systems.com/articles/print/volume-20/issue-6/features/understanding-laser-based-3d-triangulation-methods.html>.

Urban pavlovcic, Matija jezersek, Janez mozina. “Laser triangulation system for the measurment of volume and color of wounds.” Lugano, Switzerland: 2nd International conference on 3D body scanning technologies, 2011.

USB Accelerator. n.d. Coral (Google). 22 October 2022. <https://coral.ai/products/accelerator/>.

V. Mohan Malhotra, Nicholas J. Carino. *Handbook on Nondestructive Testing of Concrete*. CRC Press, 2003.

W. O. Ajagbe, O. S. Ojedele. “Structural investigation into the causes of cracks in building and solutions.” *American Journal of Engineering Research* 7.8 (2018): 152-160.

Wagh Rameshwar ashok, Dr. M S Panse, Hemant Apte. “Laser triangulation based object height measurement.” 2.3 (2015).

Weiguo Lin, Yichao Sun, Qiaoning Yang, Yaru Lin. “Real-time comprehensive image processing system for detecting concrete bridges crack.” *Computers and Concrete* 23.6 (2019): 445-457.

Weixing Wang, Zhiwei Ye, Wei Liu, Yanzhong Hu. “3D Laser Scanning Technique and Pattern Recognition Algorithms for Rock Fracture Measurements.” *International Workshop on Intelligent Systems and Applications, 2009. ISA 2009*. Wuhan China, 2009.

What Is LabVIEW? n.d. National Instruments. 21 October 2022. <https://www.ni.com/en-in/shop/labview.html>.

Xia, Xiaoling, Cui Xu, and Bing Nan. “Inception-v3 for flower classification.” *2nd International Conference on Image, Vision and Computing (ICIVC)*. Chengdu, China: IEEE, 2017. 783-787.

Yao, Yao, Shue-Ting Ellen Tung, and Branko Glisic. “Crack detection and characterization techniques—An overview.” *Structural Control and Health Monitoring* 21.12 (2014): 1387-1413.

Ying, Leslie, and Ezzatollah Salari. “Beamlet Transform-Based Technique for Pavement Crack Detection and Classification. Computer-Aided Civil and Infrastructure Engineering.” *Computer-Aided Civil and Infrastructure Engineering* 25.8 (2010): 572-580.

Yiqing Liu, and Justin KW Yeoh. “Robust pixel-wise concrete crack segmentation and properties retrieval using image patches.” *Automation in Construction* 123 (2021): 103535.

Yong Xia, Peng Zhang, Yi-qing Ni, and Hong-ping Zhu. “Deformation monitoring of a super-tall structure using real-time strain data.” *Engineering Structures* 67 (2014): 29-38.

Younes Hamishebahar, Hong Guan, Stephen So, and Jun Jo. “A Comprehensive Review of Deep Learning-Based Crack Detection Approaches.” *Applied Sciences* 12.3 (2022): 1374.

Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk. “Deep learning-based crack damage detection using convolutional neural networks.” *Computer-Aided Civil and Infrastructure Engineering* 32.5 (2017): 361-378.

Yuanfeng Duan, Qianyi Chen, Hongmei Zhang, Chung Bang Yun, Sikai Wu and Qi Zhu. “CNN-based damage identification method of tied-arch bridge using spatial-spectral information.” *Smart Structures and Systems* 23.5 (2019): 507-520.

Yue Fei, Kelvin CP Wang, Allen Zhang, Cheng Chen, Joshua Q. Li, Yang Liu, Guangwei Yang, and Baoxian Li. “Pixel-Level Cracking Detection on 3D Asphalt Pavement Images Through Deep-Learning-Based CrackNet-V.” *IEEE Transactions on Intelligent Transportation Systems* (2019).

Yung-An Hsieh, and Yichang James Tsai. “Machine learning for crack detection: Review and model performance comparison.” *Journal of Computing in Civil Engineering* 34.5 (2020): 04020038.

REFERENCES

Yusuke Fujita, Yoshihiro Mitani, Yoshihiko Hamamoto. “A method for crack detection on a concrete surface.” *IEEE*. Japan: The 18th International Conference on Pattern Recognition (ICPR'06), 2006. 901-904.

Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. “A survey of convolutional neural networks: analysis, applications, and prospects.” *IEEE transactions on neural networks and learning systems* (2021).

Zhang Lei, Fan Yang, Yimin Daniel Zhang, Ying Zulie Zhu. “Road crack detection using deep convolutional neural network.” *IEEE international conference on image processing (ICIP)*. Phoenix, AZ, USA: IEEE, 3708–3712. 2016.

Zhenhua Zhu, Stephanie German, and Ioannis Brilakis. “Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation.” *Automation in Construction* 20.7 (2011): 874-883.

Zhun Fan, Chong Li, Ying Chen, Jiahong Wei, Giuseppe Loprencipe, Xiaopeng Chen, and Paola Di Mascio. “Automatic crack detection on road pavements using encoder-decoder architecture.” *Materials* 13.13 (2020): 2960.

Advanced imaging techniques for damage characterization of concrete

A Thesis Submitted to
Nirma University
In Partial Fulfilment of the Requirements for
The Degree of
Doctor of Philosophy
in
Technology & Engineering

By
KAPADIA HARSH KHODIDAS
(16PTVPHDE164)

Electronics and Instrumentation Engineering Department
Institute of Technology, Nirma University
Ahmedabad, Gujarat, India
November 2022

Chapter 6 Conclusion and Future scope

Structural health monitoring of a new and old infrastructure is an inevitable need for the current urban livelihood. One of the key aspects of the monitoring is crack detection on the surface of various structures. Additionally, the monitoring of cracks and crack properties should be performed regularly at identified critical locations in infrastructures like a bridge. The crack detection and monitoring methodology presented in the report addresses this problem with the use of machine vision and DL. A thorough summary of the complete work and contributions is presented in the conclusion section. It is followed by the possible future extensions of the research work.

6.1 Conclusion

Concrete crack detection is a trivial problem owing to the inherent variations on the concrete surface, fluctuating light and image acquisition conditions. The developed methodology addressed the problem of concrete crack detection by developing an efficient and novel methodology for crack detection in standard concrete cubes. An industrial machine vision system was integrated with DL-based framework to perform accurate crack detection. Additionally, monitoring of crack development in the concrete cube was presented. The novel methodology was implemented for compression testing of concrete cubes. The methodology included the development of an image and load data acquisition setup, CNN-based crack detection framework, and real-time extraction of crack -related data like the number of cracks, length of cracks, and area of cracks with respect to the applied compressive load.

Generally, compression tests carried out on concrete cubes using a compression testing machine in the laboratory give peak load resisted by the cube. The developed machine vision system gives data about crack formation and its propagation during compression tests in addition to the peak load resisted by the cube. Crack-load analysis can be further utilized for understanding the behaviour of concrete with different constituents.

Data collected using the machine vision system can be studied after the completion of the compression test for interpretation of results which is not possible in conventional compression testing of concrete cubes. Continuous monitoring of cracks with respect to load and subsequent analysis of associated parameters can be used to assess the load-carrying capacity of the structure. Based on the crack-load analysis information, suitable actions may be taken to restore the designed strength of the structure. From the measurement of load and crack developments, resistance offered by structure in terms of stresses can also be evaluated which is useful for understanding the behaviour of structures in real-time. The CNN methodology implemented in the work focuses on accurate crack detection and neglects the non-crack regions like noises, dents, and blemishes, which gets segmented due to their similarity in the pixel intensities. The work combines the use of the CNN-based methodology for accurate crack detection along with the measurement of crack properties with respect to applied compression load.

Continuous data of load and concrete cube surface image during a standard compression test was acquired in a synchronous manner using the developed acquisition setup. The cracks occurring on the concrete cube surface under compressive load were detected accurately using a CNN-based novel strategy. The Inception-v3 model was trained to distinguish between crack and non-crack contours in an accurate manner. Performance evaluation and comparative analysis indicate the efficacy of the novel methodology. An average accuracy of 99% was observed in crack detection along with high recall, precision and F score. With the use of a CNN, concrete cracks can be accurately detected and monitored for growth with respect to applied loading. Furthermore, the detected cracks were processed using conventional image processing techniques to extract parameters like crack length and crack area. The data of crack initiation, number of cracks, crack length and crack area with respect to the applied load were also obtained with the developed methodology. These parameters were then monitored and analysed with respect to the applied compressive load to understand the concrete cube behaviour subjected to compressive loading. Interpretation of crack-load analysis data collected using the developed system gives a better understanding of the behaviour of concrete.

The crack-load analysis data obtained from the present work can be used to predict the occurrence of crack when the structure is subjected to load beyond its capacity. This limit can be estimated from the crack-load analysis carried out in the work.

The properties of the crack in terms of length, width, area and location obtained from the crack-load analysis is useful in assessment of damage to concrete. Such assessment will be further useful in selecting suitable repair method to be implemented by a civil engineer. Depending on the extent of concrete damage, civil engineer can choose grouting of cracks, providing cross reinforcement and/or wrapping of concrete surface with high strength materials. Here the scope of present work is limited to damage detection, so repair methods are not dealt with in detail.

The novel approach can also be extended for crack monitoring during the different types of loading on various concrete elements in the laboratory environment as well as for monitoring the health of concrete structures. The occurrence of cracks at critical locations in real-life concrete structures like bridges, buildings, and pavements can be monitored with the methodology. The crack-load analysis data can be further used to interpret the causes of cracking and appropriate measures can be suggested to strengthen the structure. The machine vision system can perform repetitive and monotonous tasks of concrete structure health monitoring. It can automate the entire process of manual inspection and health monitoring. Engineers will benefit from the integrated solution of accurate crack detection and comparative crack-load analysis of different concrete grades in structural health monitoring.

6.2 Future scope

The present work is focussed on the crack detection and monitoring of standard concrete cubes of size 150 mm × 150 mm × 150 mm. However, the strategy can also be extended to crack monitoring on other concrete elements in a laboratory or identified critical locations in existing infrastructure. In the presented work, one surface of the concrete cube under compressive loading was observed for crack detection and monitoring. The compression testing machine frame can be modified and multiple vision systems can be placed to enable crack detection on more surfaces of the concrete cube under testing.

The presented novel methodology employs a vision system which can detect surface cracks. However, the cracking may be developed due to internal stresses caused by continuous loading. Continuous observation and detection of internal cracks and developed stresses can also be addressed in the future. Identification of the reason for the cause of crack occurrence based on its orientation or inclination in the structural member is not considered in the current scope of work. However, the challenging task can be considered in future work.

The crack-load monitoring data can be used to predict crack initiation load and the safe compressive strength of the concrete cube and the mix design. The knowledge can be used for the structural design and construction of new infrastructures. The information and observation can also be useful in the assessment of old structures and identify the need for repair. The present work is focussed on detection of surface cracks and their properties along with load in real-time. Data of cracks will help in assessment of stresses and mechanical properties of concrete. However, detection of internal cracks and assessment of stress is not attempted in present work. The cracks on concrete surface gives some idea about the extent and location of internal damage. Internal crack detection methods based on ultrasonics, thermography, strain sensors, fiber optic, and various other methods are proposed in the literature for different concrete structures. Ultrasonic Pulse Velocity (UPV) method in conjunction of present system can be used to assess the internal damage in concrete. From the present system, location for possible internal damage can be identified from surface cracks and subsequently using UPV method, the extent of internal damage can be assessed. This work can be taken up in future to extend the present work.