USMAN INSTITUTE OF TECHNOLOGY

Affiliated with NED University of Engineering & Technology, Karachi

Department of Computer Science

B.S. Computer Science / Software Engineering

FINAL YEAR PROJECT REPORT

Batch-2017

CRACK DETECTION IN BUILDING STRUCTURE USING DRONE CAMERA

By

Muhammad Zeeshan Nawaz	17B-066-CS
Syeda Midhat Kamal	17B-105- CS
Syed Murtuza Abbas Zaidi	17B-100- CS
Muhammad Areeb Uddin Siddiqui	17B-088- CS
Mamoona Tariq	17B-076- CS

Supervised by

Dr. Muhammad Wasim

ST-13, Block 7, Gulshan-e-Iqbal, Abul Hasan Isphahani Road, Opposite Safari Park, P.O. Box 75300, Karachi, Pakistan. Phone: 34978274-5; 34994305; 34982476; http://www.uit.edu

Declaration

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at USMAN INSTITUTE OF TECHNOLOGY or other institutions.

Muhammad Zeeshan Nawaz (17B-066-CS):

Signature:

Syeda Midhat Kamal (17B-105-CS):

Signature:

Syed Murtuza Abbas Zaidi (17B-100-CS):

Signature:

Muhammad Areeb Uddin Siddiqui (17B-088-CS):

Signature:

Mamoona Tariq (17B-076-CS):

Signature:

Date: 14th January, 2021

Acknowledgments

It would be our pleasure as we tend to express our sincere gratitude to all the individuals involved in the accomplishment of the project. First and foremost, Prof. Dr. Zahir Ali Syed (Director UIT) and Engr. Parkash Lohana (Head of Department) for provision of facilities that assisted in the successful completion of the project. Our heartiest gratitude to project supervisors Dr. Muhammad Wasim, and Dr. Lubaid Ahmed for guiding us throughout the process and rendering invaluable help. Furthermore, our cordial gratitude to Dr. Abdul Aziz (External Supervisor) for assisting us through your extensive knowledge of the domain. We would like to thank our family members for being the source of inspiration. An appreciation for all the group members as the project would not have attained the form without their cooperation and efforts.

Abstract

The process involving crack detection is regarded to be a crucial task with reference to ensuring the safety of the structure as well as monitoring the health of the structure. Cracks are visually inspected by workers but the crack detection process which is manually done by the inspectors is painstaking, costly, and time-consuming. There are different methods to detect cracks in a building but these methods are difficult and expensive to implement. In this project, drone is used for capturing the building crack images and utilize a crack detection strategy involving image preprocessing and algorithm for the efficient crack detection. The proposed model of determining the severity of the cracks is dependent upon the decision parameters namely ratio and mean. The severity of the cracks is based on some parameters as negligible if less than a mm, slight 1 to 5 mm, moderate 5 to 15 and severe if up to 25 mm wide. The goal of this method is to provide enhanced and accurate results for the process of crack detection using a drone camera for the assistance of the building inspectors.

Keywords: Image Processing, Crack Detection, Drone Camera

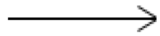
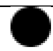
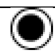
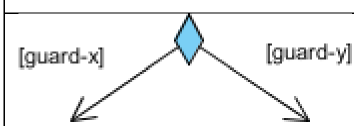
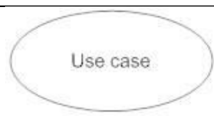
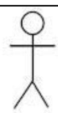
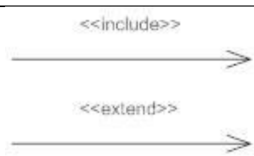
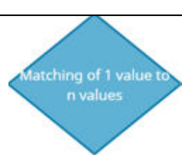
Table of contents

Contents

List of tables

List of figures

List of symbols and Units

Symbols	Description	Reference to Diagram
	**Action:** Represent a task to be performed	4.3
	**Object Flow:** The following arrows represents the flow between the different activities or actions	4.3
	**Initial Node**: Represent the initiation of any activities or actions	4.3
	**Final Node**: Represent the ending of any activity or control flows which were occurring in the activity	4.3
	**Decision Node:** Represent a decision box which tells that on which path the control will flow	4.3
	**Use cases**: It represents .the variety of uses which the user can take	4.1
	**Actors**: These are representation of the people which are making use of the use case defined above	4.1
	Associations: It is the relationship between the actor and the cases.	4.1
	**Include relationship**: It is just like the associations but depends upon the behavior.	4.1
	**Start/End:** It indicates the beginning and ending of the process.	6.1
	**Process Box:** It identifies the occurrence of a process within the system.	6.1
	**Decision Box:** It shows that some kind of decision is occurring within system. It has 2 arrows associated with it namely Yes and No.	6.1

1. Introduction

In the current development of infrastructures using concrete, the durability of such structures play an important role for the people with regard to their safety. On the other hand, cracks are considered as possible experienced infection for the concrete structures. A process for the maintenance and systematic detection is required to guarantee the durability of the concrete structure. Detection of surface cracks is a significant process in observing the underlying condition of concrete structures. If cracks develop and continue to propagate, they decrease the powerful burden-bearing surface region and can cause degradation of the structure over time.

The manual inspection of crack detection in concrete buildings is a time-consuming as well as a hectic task for the inspectors as it is difficult for them to reach high-rise buildings and detect cracks severity. In the manual inspection, the irregularities are manually noted by inspecting the sketch of the crack. The presence of cracks depict the need for repairing building; therefore, crack detection is considered as an extremely important procedure for ensuring the safety and durability of the structures. Moreover, negligence in this regard could lead to the degradation of building structures resulting in loss of lives and property. The project is intended for the assistance of the building inspectors and the domain of the project will be disaster management. Different researches have been conducted on detecting the cracks in concrete bridges, tiles, and other surfaces; however, the assistance of drone technology has not been taken to aid the inspectors. The drone is an assistance for the inspectors to inspect and confirm the cracks in high-rise buildings efficiently. Moreover, confirmation from two resources increases the efficiency of the procedure.

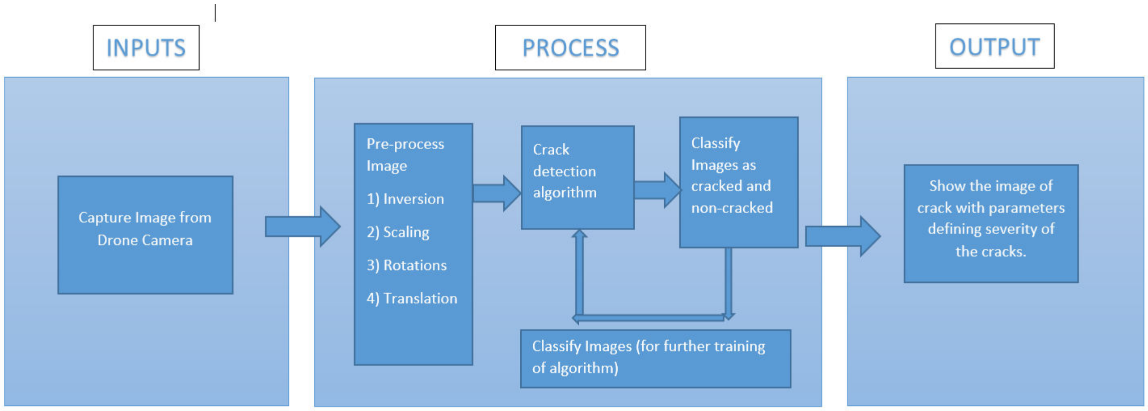


FIGURE 1.1: SYSTEM DIAGRAM

The basic outlay of the chapters is as follows: Chapter 2 - The Literature Review will provide a brief background of problem in the perspective of this project. Literature review would shed light on the previous work done related to this project. Different research has been carried out related to the crack detection. Secondly, Chapter 3 - Aims and Statement of Problem will explain the main objective of this project by giving the proper information about previous or manual methods which were used for the crack detection. A succinct summary of the problem within the context of stakeholders and technologies will be also be dealt in this chapter. Chapter 4 – Hardware, Software Analysis and Requirement will render information regarding hardware and software that were used in this project. It will give the descriptive information about the equipment that has been used to make the project work decently. It will also provide details about the software applications used to design this project along with the feature list and non-functional requirements. Chapter 5 – Software Design and Modelling provides an overview of the system architecture through help of different UML diagrams and also discusses the flow of application. Chapter 6 – Algorithm Analysis and Complexity presents the proposed algorithm along with its time and space complexity as well as comparison with the other similar algorithm approaches. Chapter 7 – Implementation provides the core code of the crack detection along with an overview of the project and other UML diagrams such as Component, Operational, and Deployment. Moreover, it also explains the HCI elements taken into account during designing GUI using MATLAB. Chapter 8 – Testing is focused upon the white and black box testing of the project. Chapter 9 – Conclusion provides an overview of the implementation, whereas, Chapter 10 – Future enhancements and Chapter 11 - Achievements discusses the scope of further expansion of the project in the future and to what extent the project was successfully achieved respectively.

2. Background and Literature Review

2.1 Introduction:

The need for an automated crack detection system evidently arises due to the immense difficulty faced by the inspectors in inspecting the different dimensions of cracks on the concrete building structures. Several research works have been previously carried out to develop an effective crack detection system for different concrete structures including pavements, bridges, roads, and buildings. These systems varied due to differences in the domain, algorithms, and techniques used for edge detection, pre-processing, and crack detection process. Furthermore, some researchers made use of a large data set for their models while some focused on model training through a small-data set. The following chapter will effectively conduct a comparison between the different techniques and algorithms used by the researchers in different domains such as machine learning and computer vision.

2.2 Importance of Crack Detection

According to [1], the process of detecting surface cracks in the concrete structures is an integral part of monitoring the structural health of such structures. The loading bearing surface area tends to reduce due to the development and propagation of the cracks. Usually, the manual inspection process is being conducted for crack detection. Due to the fact that, it is difficult for the inspectors to reach high-rise buildings and detect cracks severity, the process of manual crack detection in the concrete buildings is regarded as time consuming [2]. Furthermore, it is also affected by the subjective judgments presented by the different inspectors. In the manual inspection, the irregularities are manually noted by inspecting the sketch of the crack [3]. Detection of crack is an extremely necessary procedure for ensure the safety and the durability of structure as cracks depict the need for repair priorities [4].

The structure squares of Crack are: 1) an edge location calculation; 2) morphological activities that fill breaks; 3) tasks that recognize and separate breaks from surface deformities [31]. Moreover, negligence in this regard could lead to the degradation of building structures resulting in loss of lives and property.

2.3 Process of acquiring images:

In the research conducted by [1], a total of 20,000 concretes structures images having cracks and without cracks were being obtained from dataset. This given dataset is 227 x 227 pixels RGB image. The resolutions of these images were 6016 x 4000 pixels. The research paper by [3] represents a comprehensive overview of crack detection using drone camera is image processing technique. It mentions the difficulty associated with the process is dependent upon the image size. The data by [4] was gathered from a NIKON D7200 camera place at a shooting distance of 3-7 meters.

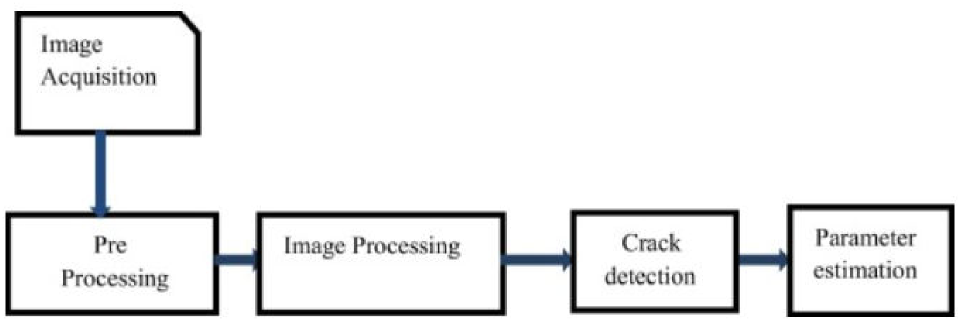


FIGURE 2.1: IMAGE PROCESSING ARCHITECTURE

Source: Mohan & Poobal, 2018

Figure 1 above provides the complete workflow of the process presented by [3]. Drone camera can also be used for capturing the images [33]. After the acquisition of the image, it goes through the process of pre-processing followed by the procedure of crack detection through using a technique. Once the crack is successfully detected, the parameters are being estimated for setting the efficiency of the system. The paper also mentions the different techniques that have been used for crack detection in different types of images such as laser images, Ultrasonic images, camera images, and IR images. The camera-based image processing techniques include threshold method for segmentation, neural networks, image stitching algorithm, filtering techniques, whereas, the IR image processing techniques involved IR thermography method and spectacular reflection. Research was also conducted by [5] where different algorithms were being tested on the images gathered from the unmanned aerial system since it claimed that the different datasets involved the crack and non-crack images which were quite smaller than the inspected infrastructures.

2.4 Crack Detection Image Processing Techniques

In the paper, [6] discussed the method of MCrack-TLS in the paper which is the image processing is combined with the terrestrial laser scanning (TLS) for improvising the transmission of the images. Another paper by [32] suggested that MATLAB is suitable for building the YOLO_v2. However, cracked images often face the effect of robust processing. The robust processing effect was optimized by the researcher [7] by using the crack characteristics and multi-scale morphological enhancement. However, the image edge that was generated after the implementation of multi-scale morphological enhancement leads to distortion. In the research by [8], it was discussed that the accurate width of the concrete crack can be detected by using the method based on the sub-pixel and optical flow. The image acquisition environment of the cracks in the concrete structures tends to be affected by the uneven illumination in the segmentation as well as potholes and stain. Therefore, it is quite difficult to implement some of the crack processing methods during the process of detecting cracks on concrete bridges. However, the research proposed by [9] was aimed at dividing the complete process scheme into three modules namely image processing followed by crack edge detection as mentioned in the paper [34], and lastly, performing the feature information extraction.

Since the implementation steps of the processing technique proposed by [9] are concerned, after the gray-scaling of the original image, the primary adaptive filtering was used many times on the image along with contrast enhancement for achieving the optimal denoising effect. Moreover, the window size was adjusted during the process initiating from 2×2, 3×3, and 4×4. In order to accurately extract the image, the improved threshold Otsu algorithm was used which was proposed in paper [36] and [37].

However, the detection accuracy was further improved by using Sobel edge gradient detection along with Otsu. Afterward, recognize and classify of the cracks are conducted through the vertical and horizontal projection of image matrix. Moreover, the data of crack pixels was also obtained for 10 crack images. The research also shows that it is also feasible for the real-time detection of the cracks in the concrete bridge structures. Figure 2 presents a complete summary of the crack detection process defined prior. The process is divided into sub-processes that are pre-processing of images, edge detection, and extraction of feature information.

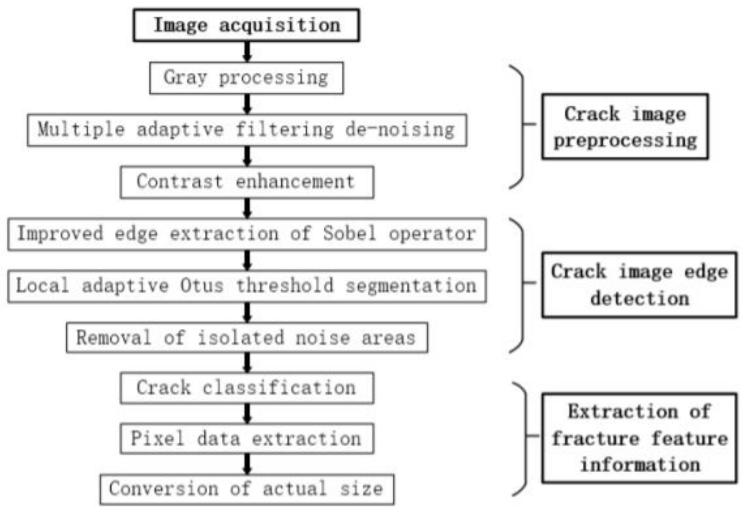


FIGURE 2.2: DETECTION OF CONCRETE BRIDGE CRACKS

Source: Wang et.al. 2019

The summary of the results indicated the effectiveness of the method in the detection of the surface bridge cracks. It shows that the absolute error between the maximum pixel width and the actual measured value was controlled within the range of 0.02mm. In the research, [2] also proposed the same Otsu method for the thresholding in the research paper. However, the Min-Max Gray Level Discrimination (M2GLD) was used for the pre-processing of the images. The main idea in this Otsu technique is that the pixels of the image are separated into two groups. However, due to the inability of the Otsu method to deliver quality results on the images subjected to blemishes and illuminations. The building surfaces are normally affected by the noise and low contrast; therefore, the M2GLD technique is used for image preprocessing.

Previously, multiple image processing techniques have been proposed from automatic image detect by the computer vision community. Some of these include edge detection, machine learning, wavelet transforms, and adaptive thresholding. According to [10], adaptive thresholding is a technique that divides the cracks into pixel-level images based on the features and their pixel values. This process helps in the simplification of the further processes. Despite the easy nature of these methods, the diversity in the cracks surface, background noises, and uneven cracks have increased the frequency of research works in the computer vision community. Another common technique used for the detection of surface cracks in the domain of deep learning is Convolution Neural Network (CNN). It is a pre-trained feed-forward network that pertains to the characteristics of parameter sharing and local connectivity which assists in making it perform better in image processing. However, according to the discussions by [1], some image augmentations needs to be done for improving the accuracy of the neural network that will be used for training the model. These image augmentations include different transformation techniques such as the vertical and horizontal flips, random color jitter for illumination, and rotations in different directions. Furthermore, it made use of the Resnet 50 model that was already trained on the ImageNet and the training of the model was conducted by using transfer learning. After the training and validation of the sample data, the model was also tested on the real-world images. Just like the previous model, [11] also made use of the Deep Learning Convolutional Neural Network where the ConvNet model was being trained with the square image patches. The significance of this study is that the dataset i.e. pavement images were gathered through smartphones and the researcher compared the effectiveness of the algorithm with the Support Vector Machine (SVM). The research by [12] was aimed to make an efficient model for the concrete roads crack detection through applying an end-to-end object detection for the purpose of training the model. Furthermore, a large amount of dataset was acquired for improving the accuracy of the model. The types of damage that was found after the application of the detection model were being classified into eight different types. Another research was conducted were 60,000 images of the concrete bridge were captured through a charged couple-device line-scan cameras. These cameras were mounted on the vehicles. In order to detect the cracks with maximum accuracy, the transfer learning method was used where an improved version of classical VGG-16 was used. The improvements were made by using Softmax classification layer and rectified linear unit for the activation function. The results obtained depicted that the model was capable of efficient feature extracted from the images [13]. However, still the CNN requires training on a larger dataset in order to ensure accurate results which makes it a drawback of the system. In the paper, [14] also discussed the EfficientNetB0 as the transfer learning method. They obtained the dataset from different previously gathered datasets. However, the image augmentation such as flip, shift, and rotations was being conducted using the built-in ImageDataGenerator interface. The Swish activation function was used in EfficientNetB0 whose architecture is shown in the Figure 3 below whereas the MBConv's framework in figure 4. The image is transferred multiple times through the MBConv's framework. This transfer learning method tends to make use of fewer parameters but provide higher accuracy.

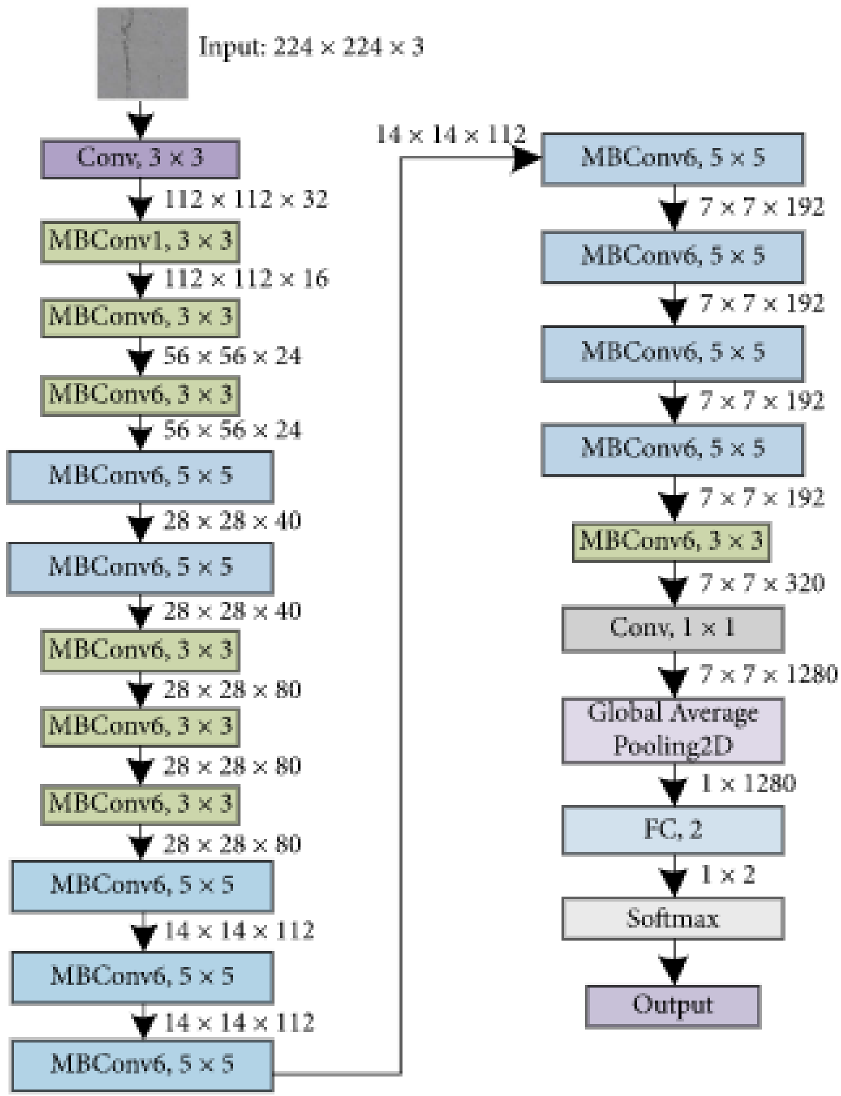


FIGURE 2.3: ARCHITECTURE OF EFFICIENTNETB0

Source: Su & Wang, 2020

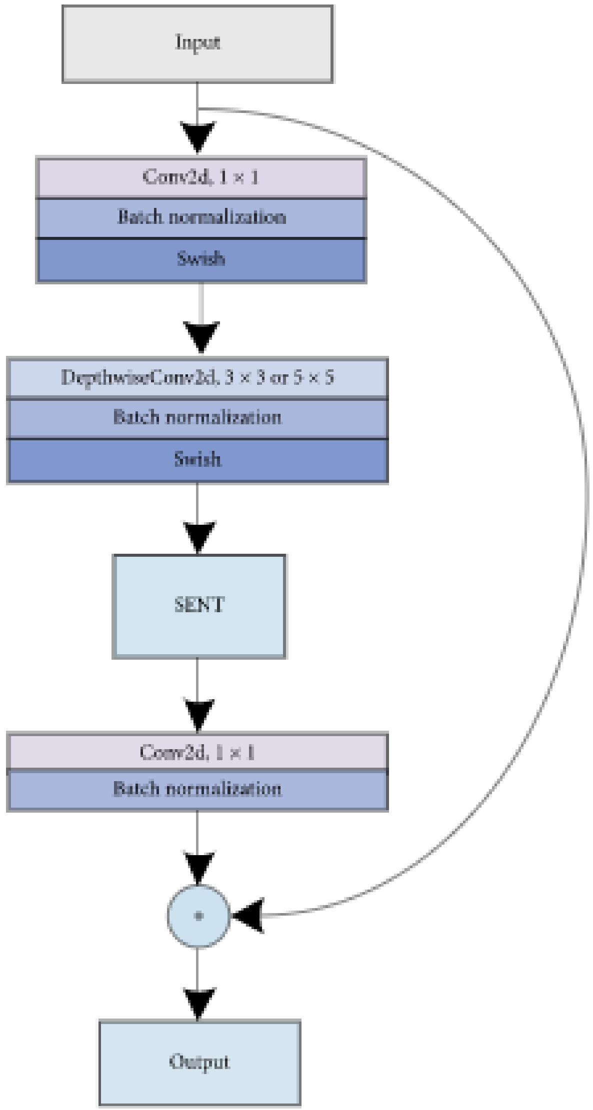


FIGURE 2.4: MBCONV'S FRAMEWORK

Source: Su & Wang, 2020

Another kind of V-shaped detector method was proposed by [4] for detecting the V-shaped features. It also claimed that the cracks of 0.2mm or even less than that can be easily detected with high accuracy using this method. The process flow of this method initiated with the gray scaling of the images, and noise processing including edge detection. The edge detection is conducted using the V-shaped detector where the image is scanned horizontally to calculate the differences in the values of the pixels. Afterward, the noise removal is done using pre-defined line definitions which check for "linearly similar" edges.

As far as the images obtained by the unmanned aerial system are concerned, the crack detection algorithms involve the emphasis on the edges through the application of filters. The research conducted by [5] involved an effective analysis of 6 filters that could be used for edge detection which were Laplacian of Gaussian, Roberts, Sobel, Gaussian, Prewitt, and Butterworth. The gray-scale conversion has to be applied to the image and then the filter is applied in the spatial domain, whereas, in the case of the frequency domain, some additional steps are required. The analysis was also conducted on the basis of precision, processing time, accuracy, and minimum detectable width. It was found that the Laplacian of Gaussian filter yielded the best output in terms of precision (88%), processing time (1.18s), accuracy (92%), and best minimum detectable width. Another research by [18] suggested that the computer vision techniques were used for the pavement crack detection; however, the major drawback of such techniques arose during different light and texture conditions. Furthermore, the laser scanning is an expensive method for detecting the pavement cracks despite the fact that it provides higher accuracy. The research concludes that the number of layers of convolutional neural network are directly proportional to accuracy. Deep learning model is regarded to provide 98% more précised results as compared to the pixel based analysis method [19]. In the research by [20], the edge detection methods were being compared with the deep neural networks on a dataset of 3240 sub images. The results showed that around more than 53% but less than 79% of the cracked images dataset was being detected; however, one of the major drawback was noise. Furthermore, it also concluded that such methods can be efficiently used for the cracks having width more than 0.1mm. A semantic segmentation method namely FCN (Fully Convolutional Network) showed 90% precision after being applied on a dataset of 40,000 images. The crack density as well as detection was quite reasonable [21]. In one of the research conducted by [42], a deep neural network was being used namely U-net for the segmentation of the cracks. These cracked images were captured for the detection of the distresses on the airport runway. These images were captured using drone camera at different heights to check for the highway pavement conditions. In the research paper, [45] presents a brief summary of the methods that can be deployed for the crack detection. It can be through image processing comprising of threshold segmentation methods, edge detection, and region growing, whereas, on the other hand, machine learning methods involve unsupervised and unsupervised learning methods. The deep learning approaches include detection of cracks based on the segmentation of pixels, and classification. According to [44], cracks can be hazardous since even the pavement conditions can also create an impact on safety as well as comfort. They can increase the operating costs and create significant issues if not detected by the pavement manager; therefore, the pavement management system is being developed.

2.5 Crack Detection in Concrete Structures

In a research conducted, the drawbacks of cracks in concrete structures was taken into account. The massive structures for instance bridges and high-rise buildings are prone to forces which enables the propagation of the negligible cracks also. The technique of FCN (Fully Convolutional Neural Network) was proven to be highly efficient through a score of around 92%. Furthermore, the research also suggests that the detection and classification algorithms pertaining to boundary box-based category do not provide efficient results since the pattern of crack propagation is complex. In this approach, the encoder and decoder frameworks are used for the training of model [22]. The same approach of FCN was also proposed by [23] on a dataset of 40,000 images. The VGG16-based encoder was connected with the FCN encoder-decoder to achieve the outcome. It was observed that the technique produced précised results. The process of crack detection in any type of structures involves the analysis of the width, depth, and length of the cracks. This inspection can lead to biased judgments and errors. According to one of the research papers, the method of targeting in crack detection is weak in tracking location, whereas, the method of image processing also detects the surface cracks as concrete cracks [24]. The papers also suggests two more approaches for the negation of such errors namely Wavelet Transform and Digital Image Correlation. Another method of efficient crack detection was proposed by researchers which used the piezoelectric active sensing system. This system was based upon the method of energy diffusion. The paper suggests that the method can be efficiently deployed for detection of cracks in pavement slabs at airports and highway. Method was further divided into lamb wave-based and impedance based methods [25]. According to [26], the technique of terrestrial laser scanning (TLS) was used for obtaining the data which was further processed into octree data structure. The method assisted in compression as well as detecting cracks of various sizes and shapes. The detection and repair of the road cracks within the time frame is imperative to prevent degradation. The research paper proposed an automated system for detecting payment distress by using YOLO deep learning framework. The framework was tested on 9,053 images captured through mounting a smartphone on the vehicle. According to [27], MorphLink-C is also an effective technique for the defragmentation of the cracks present in the pavement. The basic processes involved in the technique are fragment grouping and connection. Overall, a research was conducted to analyze the performance of the pre-trained CNN networks such as AlexNet, RestNet101, and GoogleNet for the detection of the cracks in the buildings. These algorithms are precise and accurate to a great extent; however, their performance is based upon different factors such as size of dataset, number of epochs, and network depth [28]. Just like the other CNN approaches, another research paper was also based upon two objectives. One of them was to successfully detect the cracks with severity, while the other was to provide accessibility of the application to the workers which was achieved through AWS cloud [29]. A conceptual base research methodology provided by the research paper suggested Gray Intensity Correction Method for the crack detection. The method deployed Min Max Gray Level Differentiation (M2GLD) for the purpose of improving quality of the image, whereas, Otsu method was proposed for the testing purpose [30]. According to [41], the deep learning methodologies are also being prevalent for the detection of the cracks/distress on the pavements. The computer vision techniques have been previously used for speech recognition as well as machine translation; however, currently, it is rooting itself in the detection systems. A research by [43] was conducted on the masonry historical structures for the detection of the cracks in those structures. These structures are also prone to the degradation due to man-made activities as well as ageing. In this research, the process of crack detection was aimed to be automated using the Convolutional Neural Network (CNN). However, the research also took Support Vector Machine (SVM) and Random Forest (RF) into account. The research concluded that 86% and 74% accuracy was measured in the validation and testing stages respectively while using CNN and SVM model combined. Another fully deep CNN technique named CrackSegNet was suggested by [46] for the automated detection as well as segmentation of the cracks. This approach was focused on the pixel-wise segmentation of the crack. In another approach of crack detection, the binary panel image of the object is acquired. Afterwards, the percolation process is used to detect the edges then edge-line method is used for the detection of cracks as well as localization [47]. In the research, [48] suggested another useful technique using the Wavelet transformation for the detection of the damage. The techniques particularly used were Teager Energy operator (TEO) and Wavelet Transform. However, the results conclude that combination of such techniques perform better in the noisy conditions. Researchers also aimed to develop a model for the detection of the subway tunnel cracks. The system was based on the line scan camera. The approach used was that firstly, the crack was enhanced using morphological and frequency-domain algorithms. Secondly, multi-stage filtering algorithm was applied followed by improved seed growth.

2.6 Summary of Literature Works

The overall summary of the related works is presented in the Table 2.1. Moreover, some of the research papers aligned with this project and its research are covered in Appendix F.

TABLE 2.1: SUMMARY OF SIMILAR LITERATURE WORKS

Research paper Name	**Author**	**Year**	**Data Set Acquired**	**Implementation Details**
Detection Of Surface Cracks In Concrete Structures Using Deep Learning	Dwevedi	2019	20,000 concrete structure images were captured with cracks as well as without cracks 227 * 227 pixels RGB	~ Deep Learning: Convolution Neural Network (CNN) ~ Image Augmentations ~ Model made on Pytorch
Assessment of cracks on concrete bridges using image processing supported by laser scanning survey	Valença et.al	2017		~MCrack-TLS (Terrestrial Laser Scanning and Image Processing)
Research on Crack Detection Algorithm of the Concrete Bridge Based on Image Processing	Wang et.al	2019		~Gray-Scaling ~primary adaptive filtering multiple times ~adjusted window size ~Sobel edge detection with

				Otsu
An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection	Zhang et.al	2017		~Adaptive thresholding
Robust image-based crack detection in concrete structure using multi-scale enhancement and visual features	Liu et.al.	2017		~ multi-scale morphological enhancement
Image analysis method for crack distribution and width estimation for reinforced concrete structures	Yang et.al	2018		~ method based on the sub-pixel and optical flow for getting accurate width of crack
Concrete Cracks Detection Using Convolutional Neural Network Based on Transfer Learning	Su & Wang	2020	~12, 000 cracked and non-cracked images were obtained from dataset by Li and Zhao captured from smart phone with resolution 4160 * 3120 pixels ~2000 more images were used from the SDNET2018 dataset	~EfficientNetB0 using the transfer learning method ~Pre-trained weights on ImageNet ~For updating network parameters, Adam optimizer was used ~Image Augmentation using built-in ImageDataGenerator ~Swish activation function
Road crack detection using deep convolutional neural network	Zhang et.al,	2016	500 images collected from low cost smartphone 3264 * 2448 resolution	~supervised deep CNN ~ConvNets model trained on square image patches
Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone	Maeda et al	2018	9,053 images gathered through smartphone that was mounted on car. Images were of road damage	~convolutional neural networks ~end-to-end object detection method for model training
An Intelligent Classification Model for Surface Defects on Cement Concrete Bridges	Zhu & Song	2020	600,000 images of concrete bridge were captured from charged couple device line - scan cameras which were placed on the vehicle.	~used improved classical CNN i.e. VGG -16 (Visual geometry group network) ~rectified linear unit used as activation function ~Softmax classification layer used
Crack Detection on Concrete Surfaces Using V-shaped Features	Sato et al.	2018	~6016 * 4000 pixels images were captured through NIKON D7200 from a distance of 3-7m.	~ V-shaped features were detected using V-shaped detector ~Gray scaling and noise removal ~Edge Detection that calculates differences in values of pixels ~noise processing through using line definitions
Crack detection using image processing: A critical review and analysis	Mohan & Poobal	2018		~assessment of the different available algorithms

Detection of Surface Crack in Building Structures Using Image Processing Technique with an Improved Otsu Method for Image Thresholding	Hoang	2018		~Improved Otsu method for thresholding ~Min-Max Gray Level Discrimination for preprocessing
Benchmarking Image Processing Algorithms for Unmanned Aerial System-Assisted Crack Detection in Concrete Structures	Dorafshan et al.	2019	~50 images of defected as well as sound concrete were taken using Unmanned Aerial System	~For edge detection, 6 filters was used namely Laplacian of Gaussian, Roberts, Sobel, Gaussian, Prewitt and Butterworth ~The Laplacian of Gaussian from spatial was found to be most effective for real-time crack detection involving unmanned Aerial System

3. Aim and Statement of Problem

3.1 Introduction

In the process of manual inspection of the crack, the crack sketch is being made using paper and pen for analysis in a physical manner. Furthermore, the states of the different anomalies occurring are also being noted down. The manual process is prone to the objective behavior of the inspectors because it relies upon the experience and information provided by the expert inspectors. Thus, this project of automation is being proposed. The major aim of this project is to render help to the crack detection inspectors while inspecting high-rise buildings as this process is painstaking as well as time-consuming. The detailed discussion of problems with respect to the inspector's perspective is being discussed in this chapter.

3.2 Problems faced by Inspector:

In the process of monitoring the concrete floor, the crack map of the floor is being generated. This helps in examining some important information related to the health of the surface. Furthermore, the earlier the cracks are detected, they help in reduction of the cost as well as the renovation efforts. Currently, the humans are involved in the process of crack detection. Following problems occur in manual crack detection by inspectors:

3.2.1 Time-consuming:

Performing manual crack detection was very difficult and consume huge time to finish.

3.2.2 Accuracy:

Although manual crack detection was not easy but also unsatisfactory. There was no surety whether proper crack was detected or not due to the biasness in the perspectives.

3.2.3 Costly:

The cost was another issue, which occurred in manual crack detection. Inspectors had to particularly purchase some tools for crack development physically and then its detection.

3.2.4 Safety issue:

The process of carrying out crack detection in high-rise buildings might cause a threat to not only the lives of people but also causing danger to the inspector's life.

3.3 Comparison between related technologies used in crack detection:

TABLE 3.1: COMPARISON BETWEEN CRACK DETECTION TECHNOLOGIES

Crack Detection in building structures using drone camera	Crack Detection Algorithm of the Concrete Bridge	Crack Detection on Concrete Surfaces Using V-shaped Features	Crack Detection in Building Structures Using Image Processing Technique
Program the drone such that it will be able to record itself the video/images from a specified distance parameter	The images [9] will be acquired through a direct images by using CCD camera.	The images will be acquired through [1] a direct images by using camera NIKON D7200.	The images will be acquired through a direct images [3] by digital camera.
Preprocessing (invert, scaling, rotations, and translation) is used for images frames in our project.	Preprocessing (grey processing, multiple adaptive filtering de noising, and contrast enhancement) is used in [9] crack detection.	V shaped detector[9] flow this process(Gray scaling process, edge detection, noise removal processing)	The algorithm for image enhancement namely Min-Max Gray Level Discrimination (M2GLD) was used [6]. It is being used as an image preprocessing step. It is being applied before using the Otsu method for the thresholding.
There are some markers could be used for tracking the cracks. Negligible (Less than a mm), Slight (1 to 5 mm), Moderate (5 to 15 mm), Severe (Up to 25 mm wide)	It was observed that an evident gray interval boundary appeared after processing image using window adaptive filter. The gray value for background was higher as compared to the target crack. Target crack gray value was between 0 and 255*0.1, whereas, the background gray value was between 255 x 0.6 and 255.	The cracks of 0.2 millimeter in width were detected [9] from a distance of 7 meters.	The parameters of the model [6]: The adjusting ratio: RA=2; The margin parameter: =0.5; The minimum numbers of pixels :Np =round (0.001 IN x IM) ,where 001 IN x IM is the image size; The threshold value of the axis ratio index :APIT =3
Showing the complete process model of the building specifying the locations with the cracks marked according to their severity.	Showing the complete process model of the bridge specifying the locations with the [9] cracks marked according to their severity.	Showing the complete process model of the [9] Concrete Surfaces specifying the locations with the cracks marked according to their severity.	The complete process model for surface cracks was shown. Moreover, the locations as well as automatic detection and analysis of cracks was also catered. The cracks were marked according to their severity [6].
Generate complete report of the building cracks.	Generate complete table of building cracks images data [9].	At the end of the [9] process, results, the cracks having size 0.2 millimeters were accurately detected.	At the end of the process, results states that the cracks were identified to some accuracy [6]. The M2GLD can help in improving the performance of the Otsu.

4. Hardware, Software analysis and requirements

4.1 Hardware:

Our project hardware are:

- Drone for capturing images
- System for running application

4.2 Drone:

Selected drone is a programmable educational drone which has the ability to be programmed with Scratch. It has an additional feature of obstacle avoidance and also alerts on low battery. The drone can be controlled through App installed on the smart phone [15]. Moreover, it has the following features:

- Electronic image stabilization helps keep your photos and videos steady. [16]
- Vision positioning system: Downward-facing camera facilitates precise hovering. [16]

A brief summary of the specifications of the selected drone camera is shown in the Table 4.1, whereas, the details related to the drone are in Appendix A.

TABLE 4.1: SPECIFICATION OF THE SELECTED DRONE CAMERA

S.no	**Feature**	**Specification**
1	Name	KOOME K800 Quadcopter
2	Processor	Intel
3	Weighs	0.08 kg
4	Programmable	Yes

4.3 System:

The required operating system for this project is a 64-bit operating system. It should have 32 GB memory, whereas, the requirement for the processor is 3.40 GHz for Desktop Application.

4.4 Software:

In software terms for image processing, both of these strategies can be used:

- MATLAB
- Python Programming

4.4.1 MATLAB:

MATLAB can be used for many applications, include machine learning and deep learning purpose, signal processing and communications, videos and image processing etc. Furthermore, MATLAB also provides Image Processing Tool Box with a variety of built-in functions for performing image processing efficiently. Therefore, MATLAB will be used for performing the image processing. In a research paper by [49], a successful surface crack detection system was developed using the artificial vision system on MATLAB.

4.4.2 Python:

Python is one of the good choices for image processing due to his popularity for scientific tasks. Furthermore, it has a lot of available image processing tools which can be used free of cost.

4.5 Requirements

4.5.1 Inputs

- As a user, I feed the images (captured by drone) into the system to check severity of cracks.
- As a drone operator, I can capture images by drone from specified parameter distance.

4.5.2 Process

- As a data analyst, first I would collect dataset from different platforms. As a data analyst, I would also concerning on taking data from good camera devices for beginning rather than capturing it from drone.

- As a system, I will work on the preprocessing of dataset which includes invert, scaling, resizing and translation.
- As a developer, I will work on algorithm for image processing.
- As a developer, I would be working drone programming to automate it.
- As a project developer, working on the desktop application would be second last step following the procedure.
- As a project manager, last step would be to merge above processes into a complete system.

4.5.3 Output

- As a user, I want not only to specify the crack in building but it also inform about severity of damage on the basis of criteria mentioned in Table 4.2.

4.5.4 Non-Functional

The glimpse of reliability for our project can be seen in that the project will make use of the drone which is designed by the MIT professionals. It will be assured that good UI of our desktop application is developed so that the user is able to use it easily without having knowledge of further details. The project is completely planned, feasible and will be developed according to the steps given above. File integrity of the project would be conserved as it has nothing to do with cyber environment. Project is aimed at increasing the accuracy of the proposed algorithm. The project would be portable so that user can take it to any public site.

4.6 Diagrams:

In the below diagram, drone and inspection through application are taken as the main actors. First, the functionalities of drone, it can be seen in Figure 4.1, initial task of drone is to identify the crack. After that, with assistance of inspector, drone will look up for proper angle to capture the data. These were three use cases defined for drone as an actor in Figure 4.1. The second actor was inspection through application, its functionality begin with uploading data in the system to be processed. After that the data would be pre-processed to level the data for further processing. Following that, crack detection algorithm will process the data to find cracks and its severity. These were all the use cases which were linked to inspection through application as an actor in Figure 4.1.

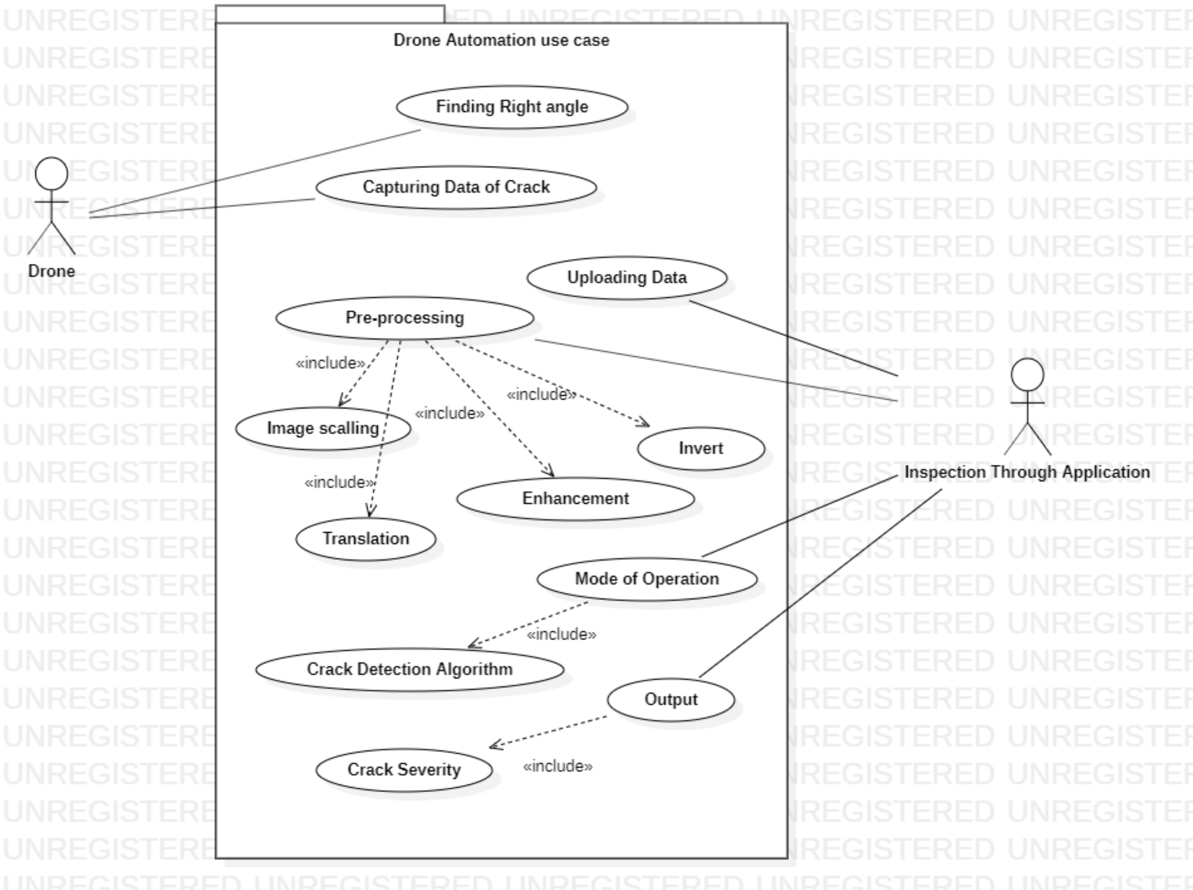


FIGURE 4.1: ACTOR USE CASE

In Figure 4.2, the drone with help of inspector will capture the data of the particular crack from different angles. The data will then by transferred from the drone to the application as shown in Figure 1.1 above. After this, the main process of the project starts which is to identify the crack and its severity from data. The first step of this process includes the pre-processing of data. Pre-processing further includes four functions:

1) Inversion
2) Scaling
3) Rotation
4) Translation

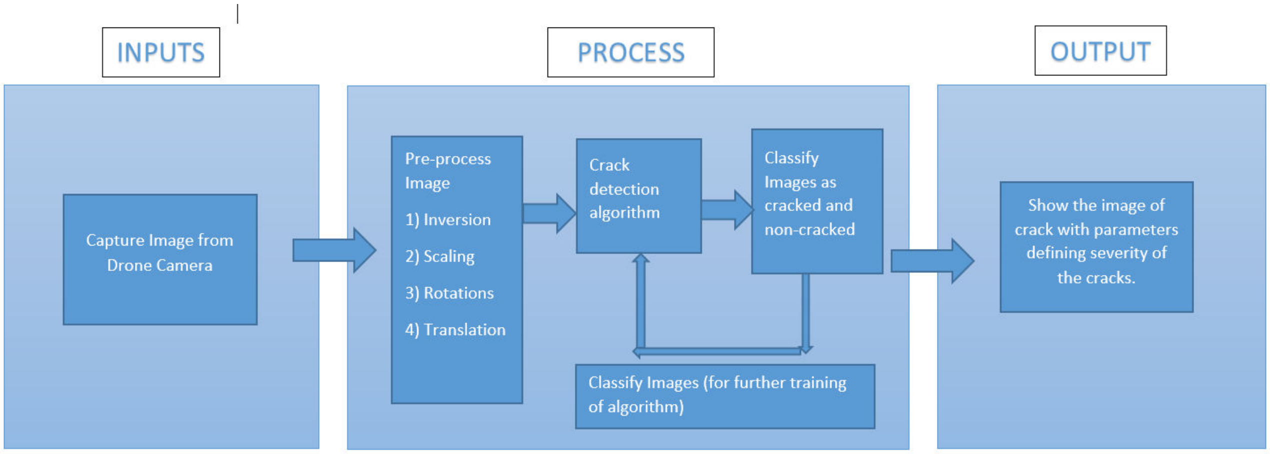


FIGURE 4.2: SYSTEM DIAGRAM

After that, the data will go through a certain detection algorithm that differentiates the dataset of images as cracked and non-cracked with respect to their intensity, width, and depth. The next process would be to detect the severity of the crack from data. Some parameters to distinguish crack severity are being defined which is given below:

TABLE 4.2: DIFFERENT TYPES OF CRACKS WIDTH

Negligible	Less than a mm
Slight	1 to 5 mm
Moderate	5 to 15 mm
Severe	Up to 25 mm wide

The output section consist of a GUI that specifies the different decision parameters for the cracks along with indicating their severity. Moreover, certain parameters would be defined for the system.

Figure 4.3 provides a clear picture of the activities occurring within the complete system of crack detection. First of all, images of the building will be acquired by using a drone. After that inspector will sign in or sign up (in case of no account) to the system if the id or password is valid then further move otherwise again sign-in option will appear. After that, some modules will be displayed on the desktop, and an inspector will select any module which he want such as view profiles or edit profile, detection of cracks which will be detected by preprocessing technique and logout.

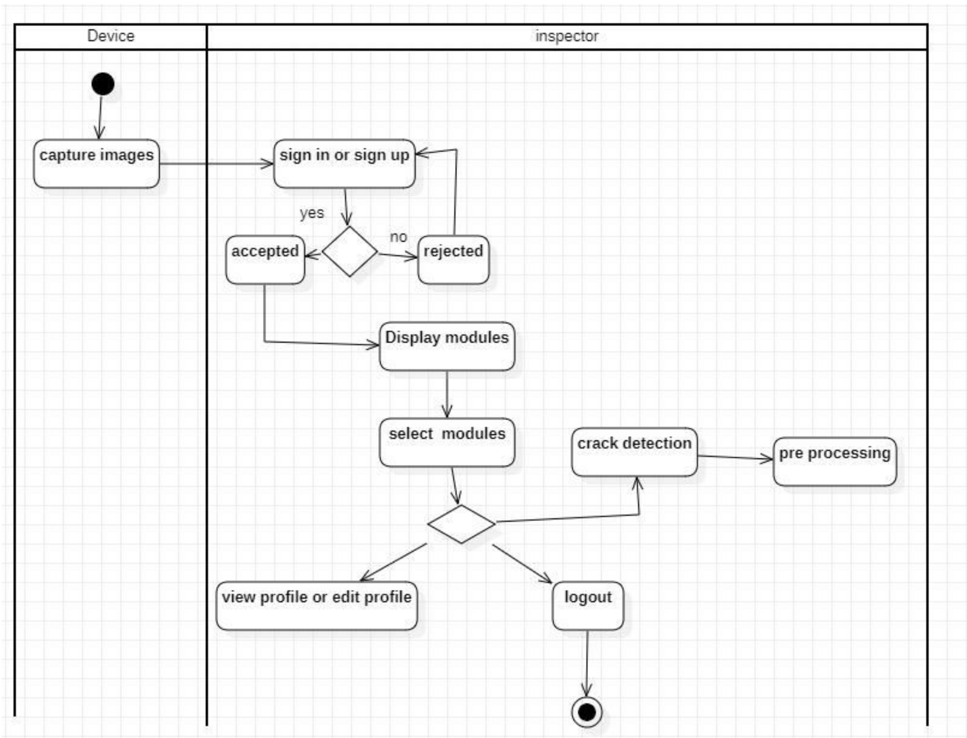


FIGURE 4.3: ACTIVITY DIAGRAM

Figure 4.4 can be briefly explained through the following

- Actors: Inspector, Drone Camera
- Processing: System Software
- Server: Database
- All relation depends upon our implementation (our system) of crack detection through image processing by using drone.

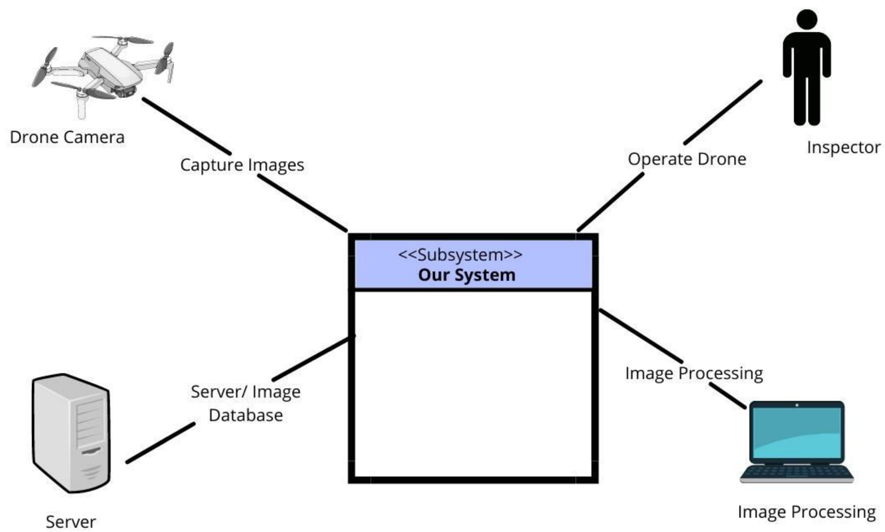


FIGURE 4.4: OPERATIONAL DIAGRAM

5. Software Design and Modelling

5.1 Introduction

The process of crack detection is labor intensive as well as time-consuming. Therefore, the project intends to develop a system where the detection of cracks and its measurements is automated to assist the inspectors in the process. The chapter namely Software Design and Modelling provides a brief overview of the project architecture with the help of different software modelling diagrams such as Use Case, Activity Diagram, and Sequence Diagram. Furthermore, the complete workflow of the desktop application interfaces made using MATLAB GUI will also be discussed in this chapter.

5.2 Project Architecture:

As far as the project architecture is concerned, the images will be captured through the camera mounted on the drone from a particular angle. These images will then be processed through the desktop application made using MATLAB GUI in order to detect the cracks after the application of the pre-processing techniques and proposed algorithm. The main concern of the crack detection algorithm is to provide the accurate as well as efficient results. The project architecture is succinctly explained through the different diagrams presented later on in the chapter.

5.3 Actor Use Case Diagram:

The actor use case diagram mainly involves 2 actors namely drone and inspector. In Figure 5.2, the functionalities of the drone is shown to be capturing the images of the cracks. In Figure 5.1, the use cases of the inspector is being shown where the user uploads the data for further processing. Then the data is being pre-processed and as per the crack detection algorithm, the crack is being detected and the system will display the original image and crack detected image along with its severity.

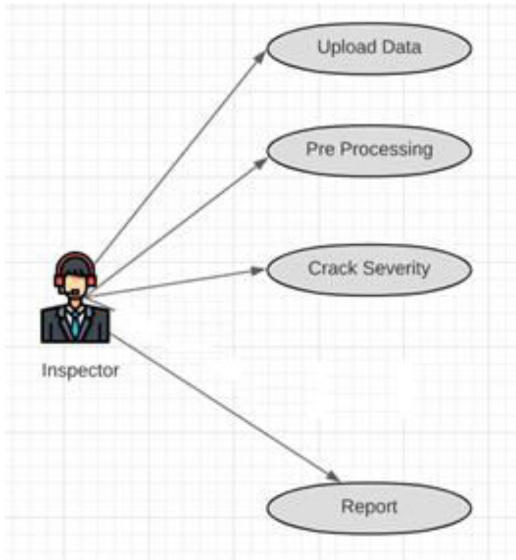


FIGURE 5.1: ACTOR USE CASE – INSPECTOR

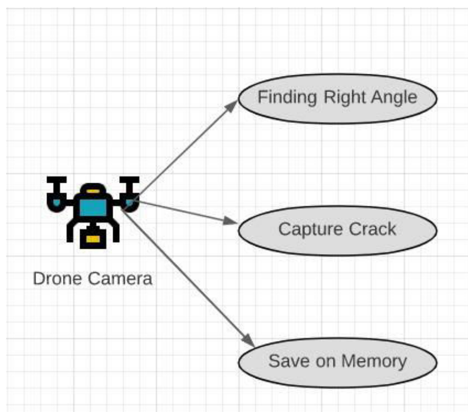


FIGURE 5.2: ACTOR USE CASE – DRONE

5.4 Activity Diagram:

The activity diagram shows all the set of the flow of project activities from one action to another action in the system. The Figure 5.3 shows the complete activity diagram of the project. In Figure 5.3, first of all, some of the images of building will be taken from a device. After that inspector will sign in or sign up (in case of no account) to the system if the id or password is valid then further move otherwise again sign-in option will appear. After that, some modules will display on the dashboard, and an inspector will select any module which he will want such as view profile, and detection of cracks which will be detected by preprocessing technique and algorithm on the backend, and then logout.

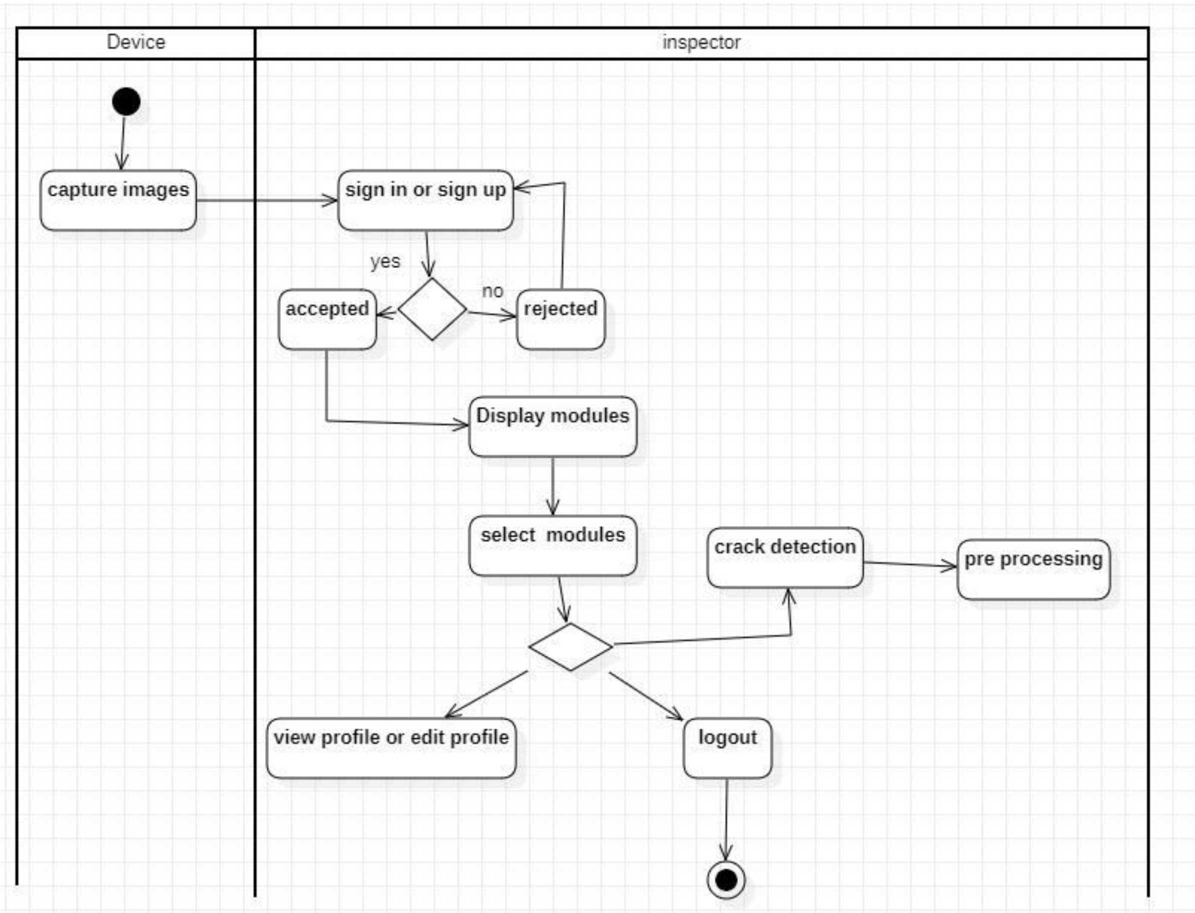


FIGURE 5.3: ACTIVITY DIAGRAM

5.5 Sequence Diagram:

The sequence diagram shows all the set of the activities of the system. The sequence diagram of the project is shown in Figure 5.4. The inspector will first capture the images and save the data in a system, the inspector will sign in or sign up the system. The system will ask for the ID and password if the match display the modules otherwise will stay on a sign-in page. After the sign in, Dashboard will appear on screen in which crack detection and profile module will show. The inspector will have an option whatever he selects, either detect the cracks or update his profile. And after that, also have a sign-out option.

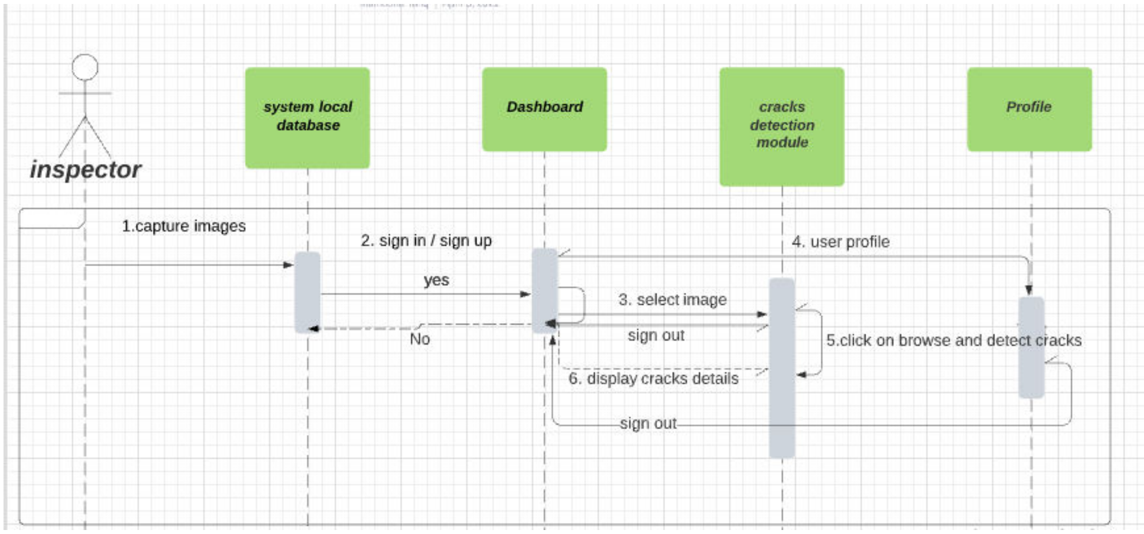


FIGURE 5.4: SEQUENCE DIAGRAM

5.6 Front-end (GUI):

Since, the application is a non-business application, therefore, all the front end GUI Interfaces are being shown. The front end GUI is developed using the MATLAB GUI. The prompts shown in Figure 5.11 and 5.13 are used for the prevention of error by the user. These errors can be going back without saving changes and logging out by mistake respectively. Apart from the dashboard GUI, Figure 5.8 and 5.12 shows the waiting prompts in order to provide efficient feedback to the user to avoid frustration and resentment during the process. It informs the user that some kind of process is carrying out.

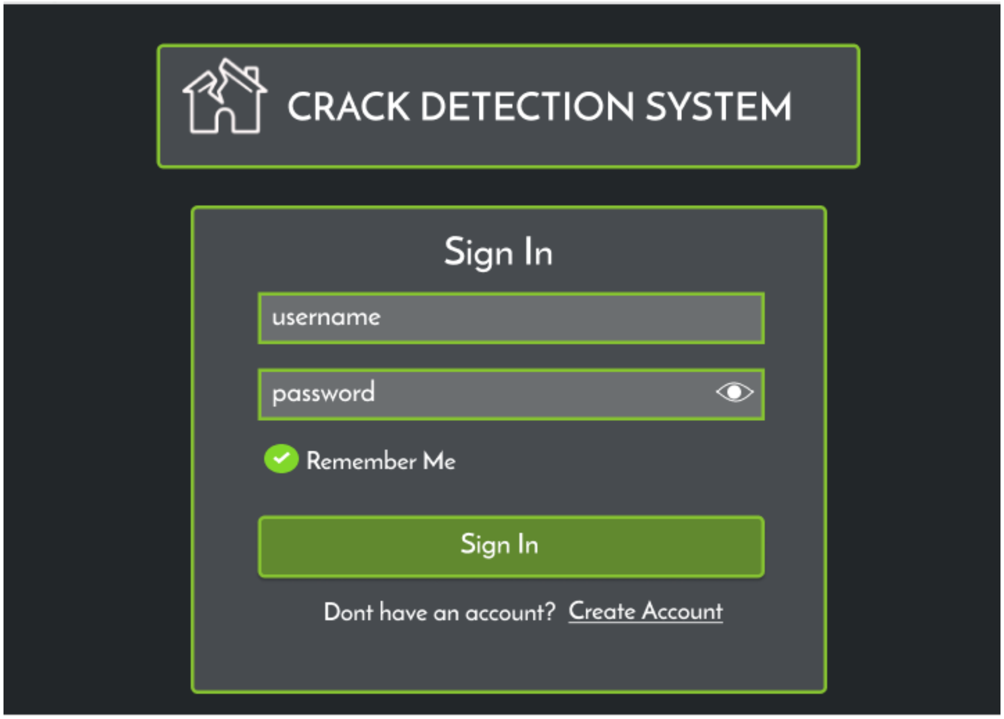


FIGURE 5.5: GUI - SIGN IN SCREEN

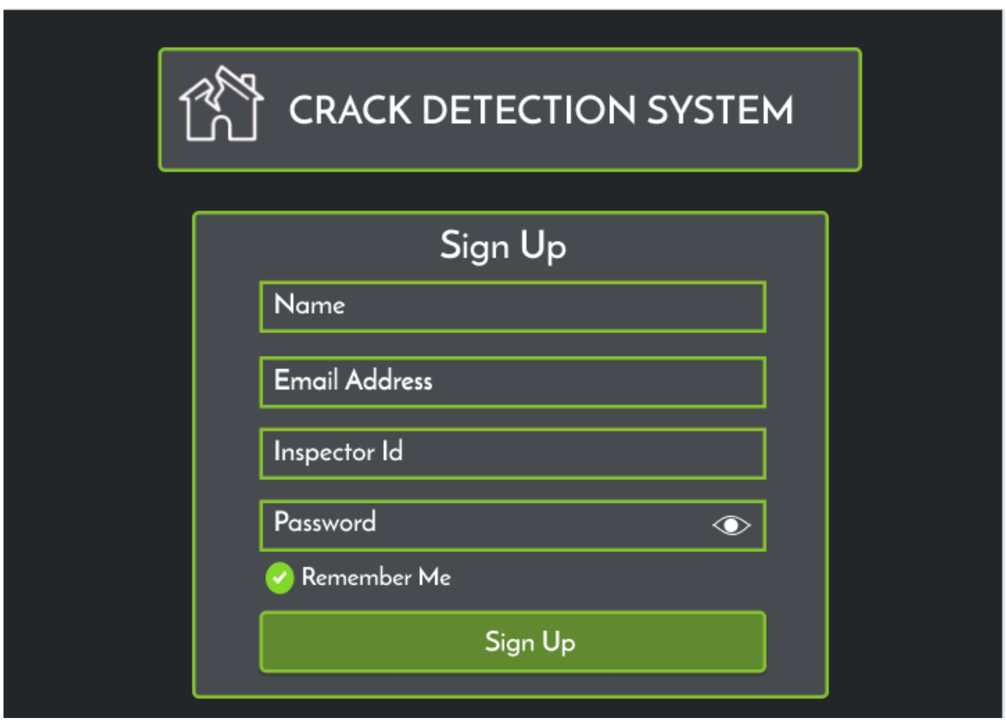


FIGURE 5.6: GUI - SIGN UP SCREEN

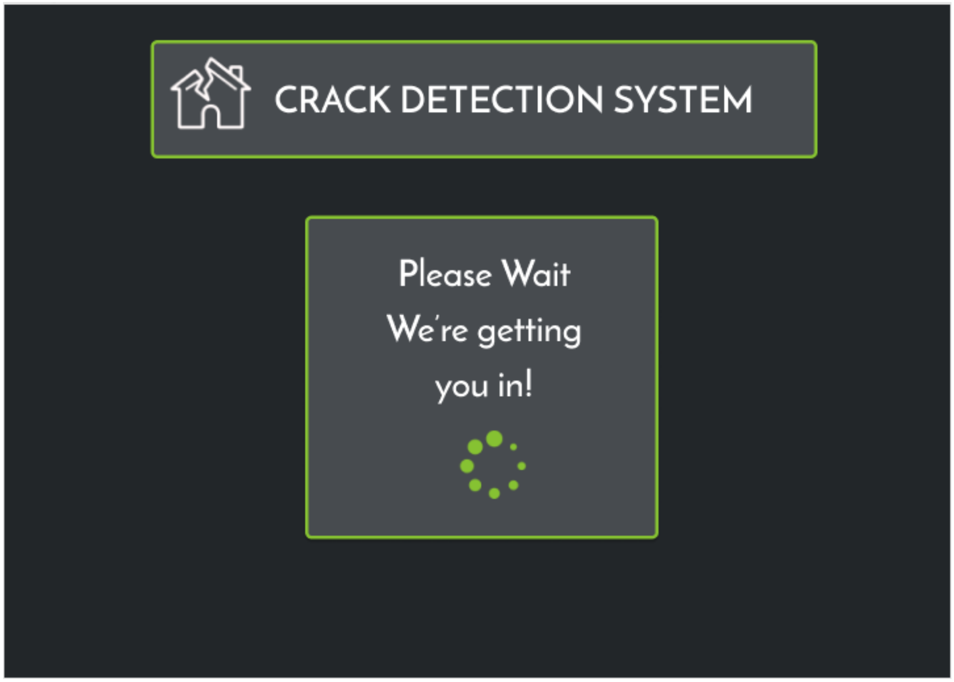


FIGURE 5.7: GUI - WAIT PROMPT SCREEN

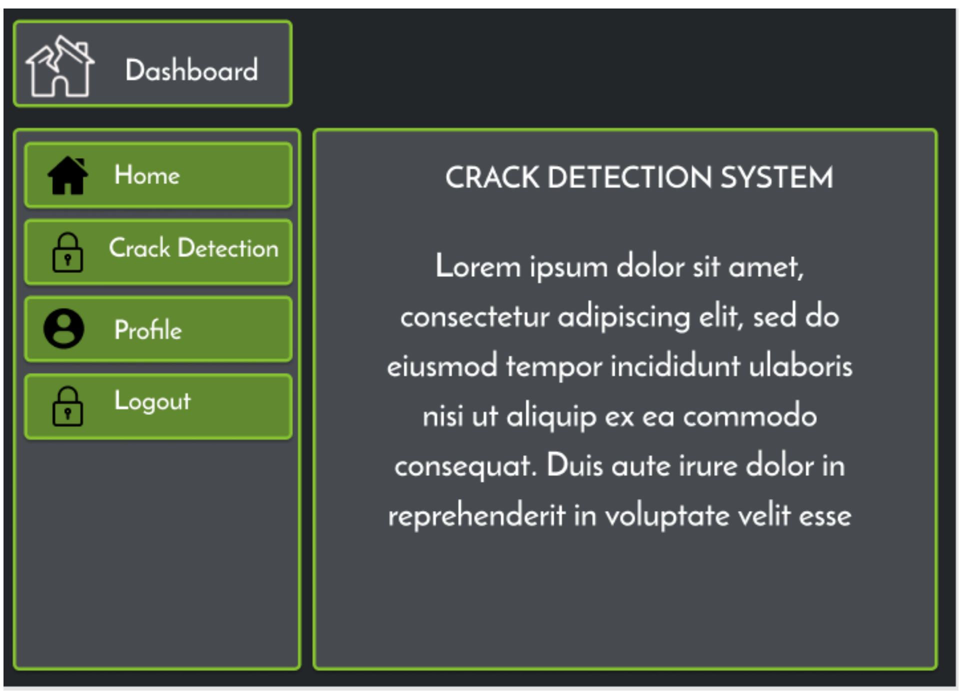


FIGURE 5.8: GUI - HOME SCREEN

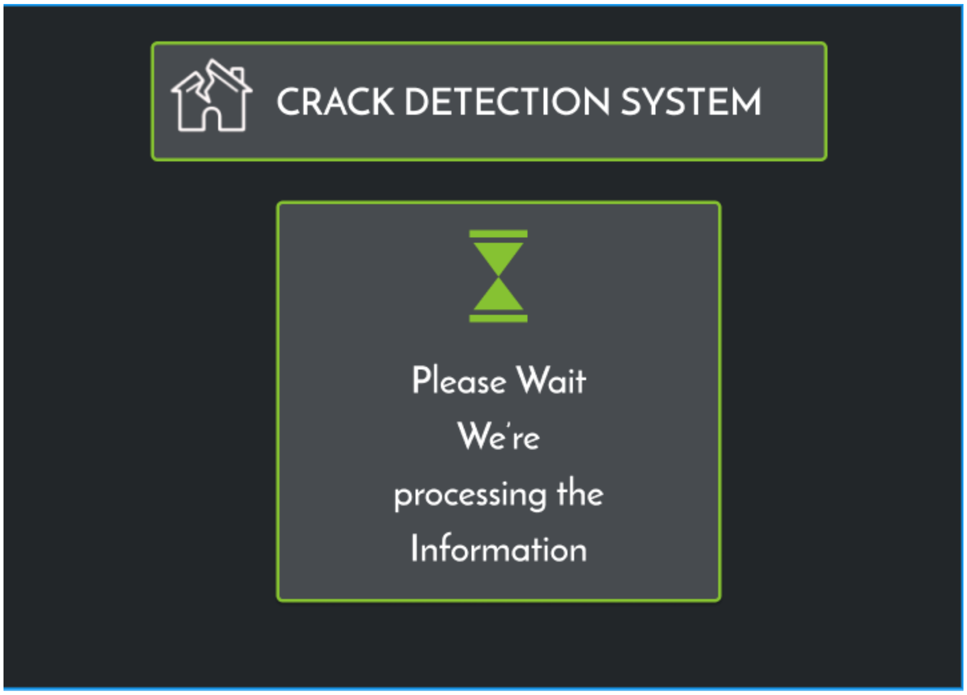


FIGURE 5.10: GUI - WAIT PROMPT SCREEN 2

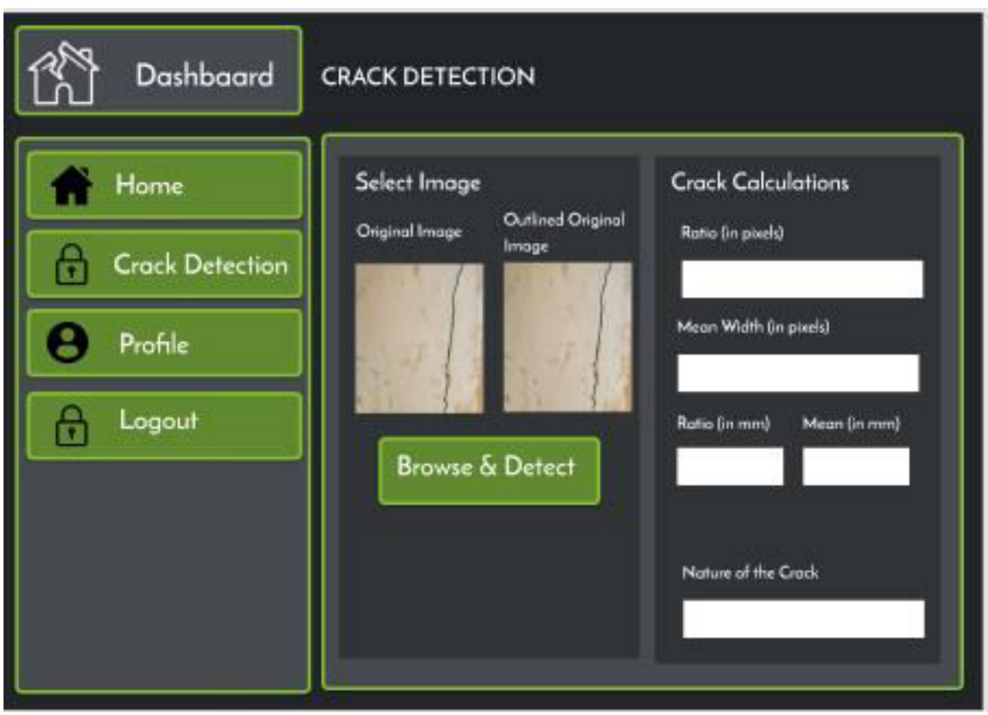
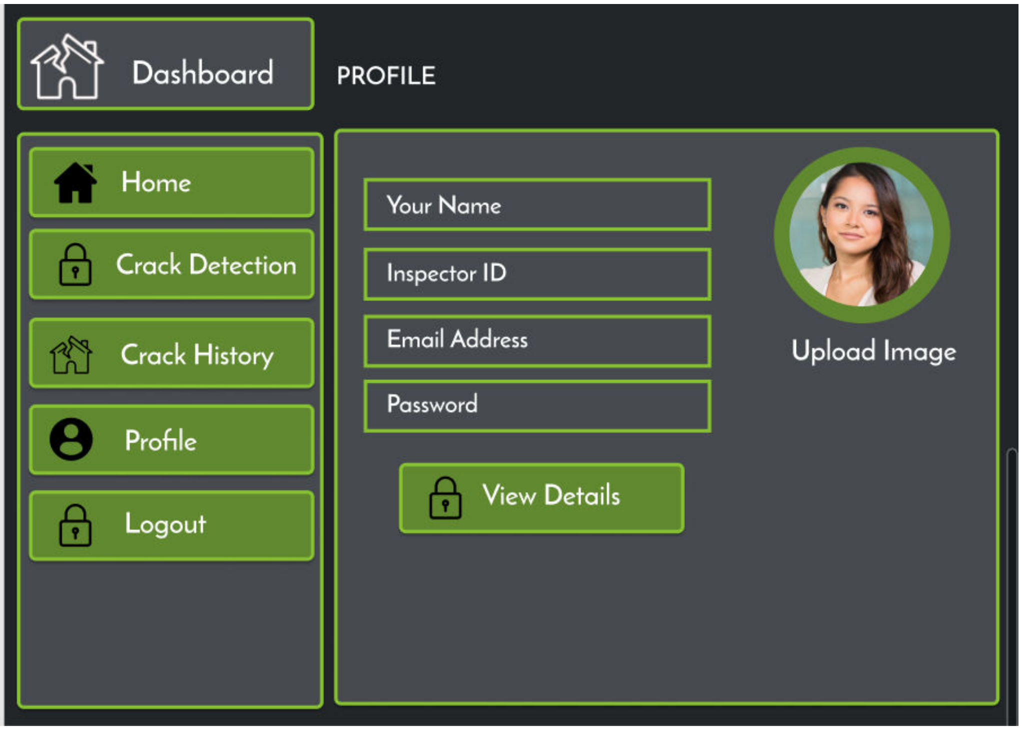


FIGURE 5.11: GUI - CRACK DETECTION SCREEN 2



]

FIGURE 5.12: GUI - USER PROFILE

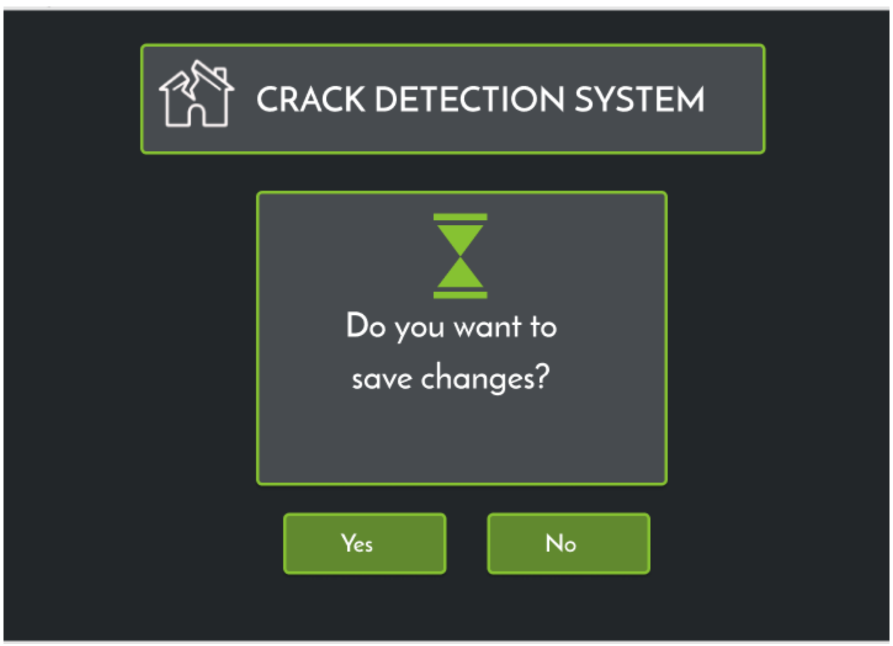


FIGURE 5.13: GUI - SAVE CHANGES ERROR PREVENTION PROMPT

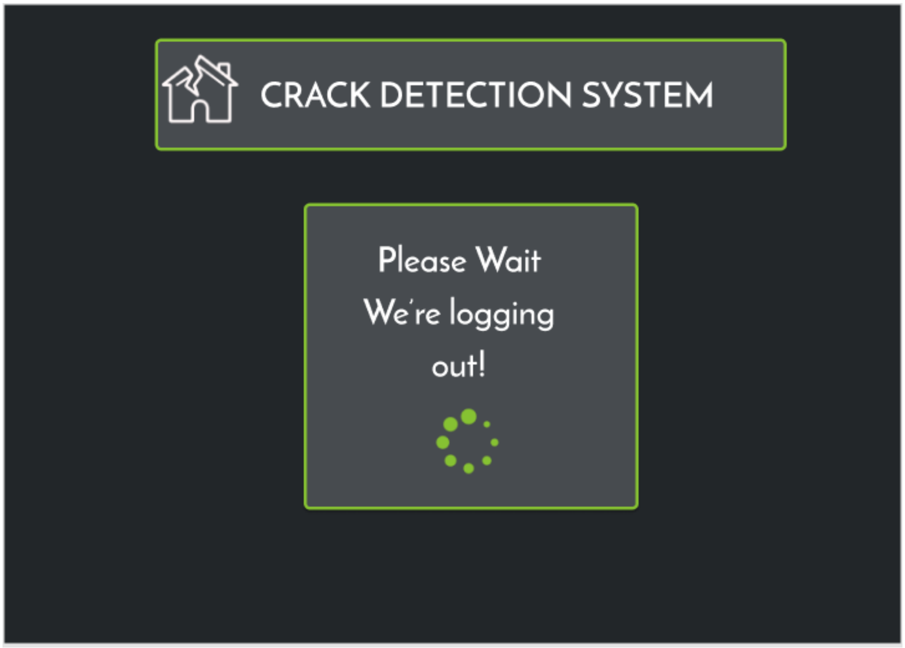


FIGURE 5.14: WAIT SCREEN FOR LOGGING OUT

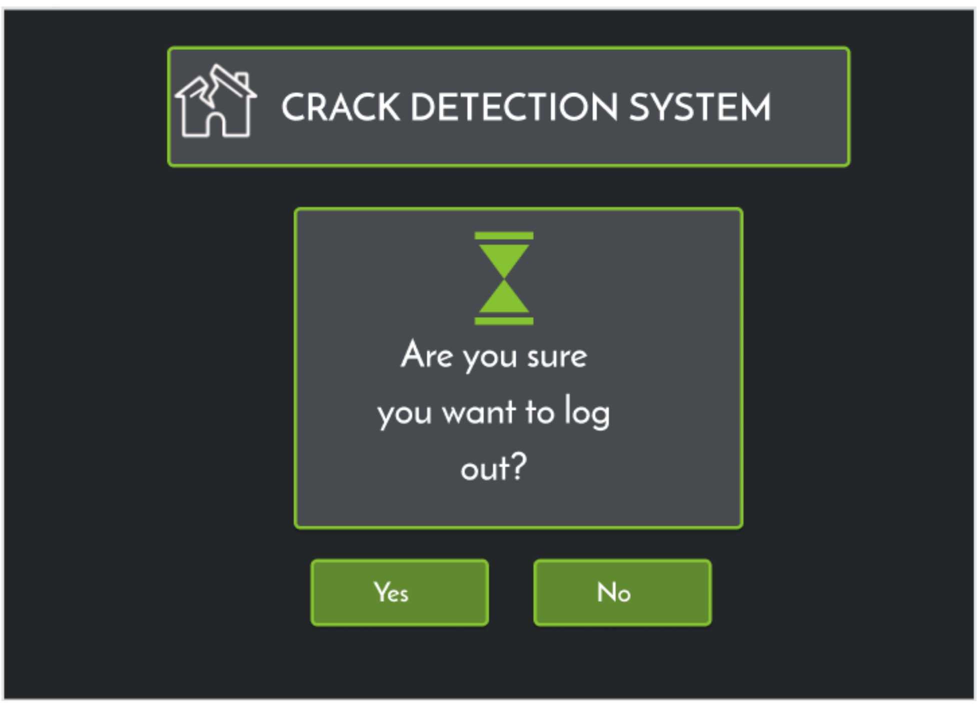


FIGURE 5.15: LOGOUT ERROR PREVENTION PROMPT

6. Algorithm Analysis and Complexity

6.1 Introduction

A variety of algorithms are being deployed for improving the efficiency of the crack detection process; however, all of them are distinct on the basis of their efficiency, time complexity, and space complexity. Different techniques are used for the detection applications, for example, Convolutional Neural Network, Machine Learning, Min Max Gray Level Discrimination, and Deep Learning. The following chapter aims to describe the algorithm developed for the crack detection process along with flowchart to assist the understanding of the algorithm. Furthermore, it also discusses the comparison of proposed algorithm with algorithms proposed in the different research works.

6.2 Proposed Algorithm

Step 1: Acquiring the images

Step 2: Crack Detection Algorithm

- Step 2.1: Basic Transformation
- Step 2.2: Converting RGB to Gray Scale
- Step 2.3: Convert into Binary based on threshold value
- Step 2.4: Application of Canny Filter
- Step 2.5: Segmentation
- Step 2.6: Decision Perimeter Calculation i.e. Mean and Ratio

Step 3: Display Cracks with calculations and Severity

The Figure 6.1 below presents a clear understanding of the complete process occurring within the algorithm. The process initiating from the Start Symbol to the segmentation of the image and calculation of the perimeter is defined as the process of registration of the crack images. The algorithm ends on the calculation of the different perimeters (Mean and Ratio) with respect to cracked images

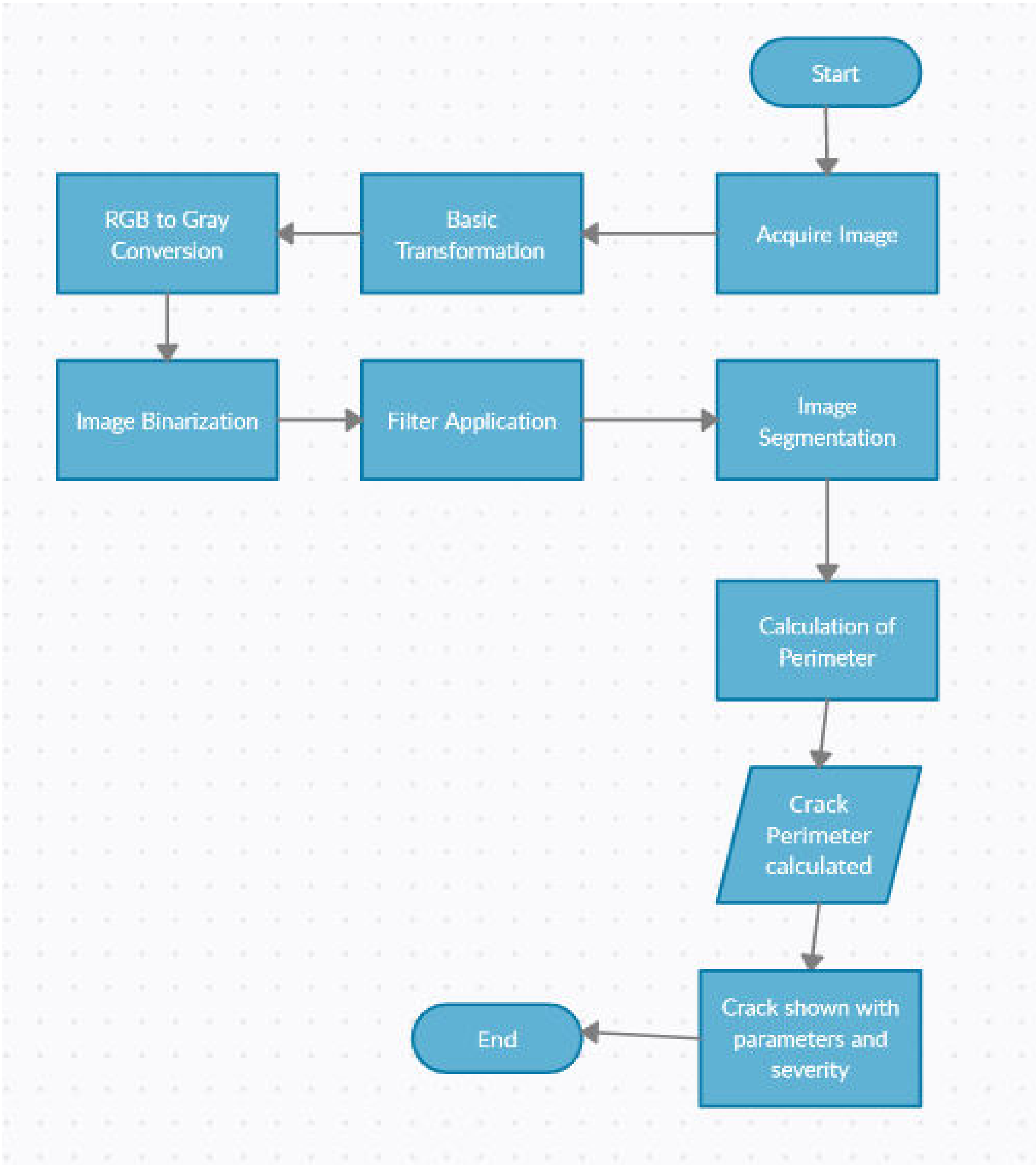


FIGURE 6.1: FLOWCHART OF THE PROPOSED ALGORITHM

6.3 Algorithm Explanation

In Step 1, the images will be acquired through using a camera mounted on the drone camera. The images from the same dataset are shown in Figure 6.2:

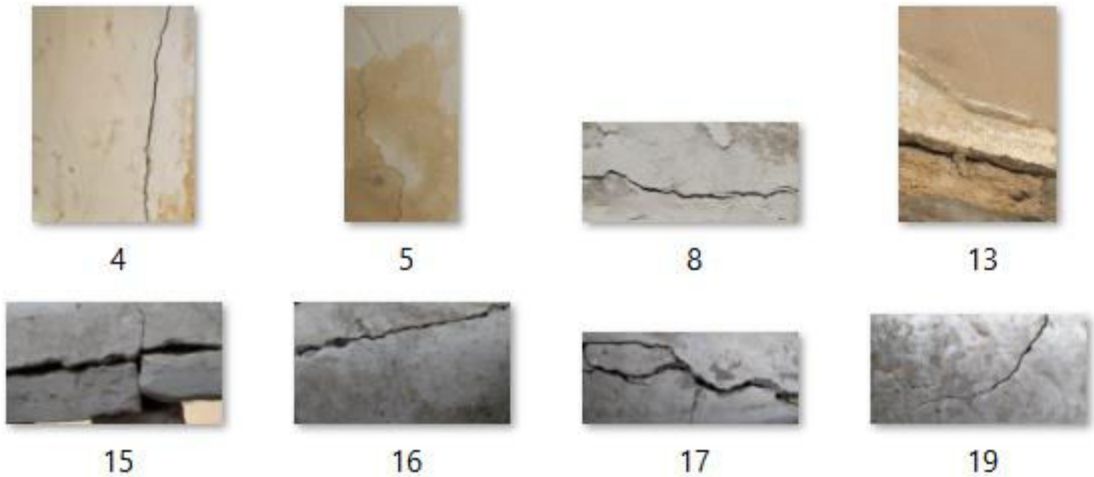


FIGURE 6.2: SAMPLE IMAGE DATASET ACQUIRED

In Step 2.1, the crack detection algorithm would be implemented consisting of different sub steps. Since the images acquired through camera have variable sizes; therefore, the operation of basic transformation will be applied on the images as shown in the Figure 6.3. In the procedure of scaling, the image size enhancement needs to be done without affecting the shape of the image.

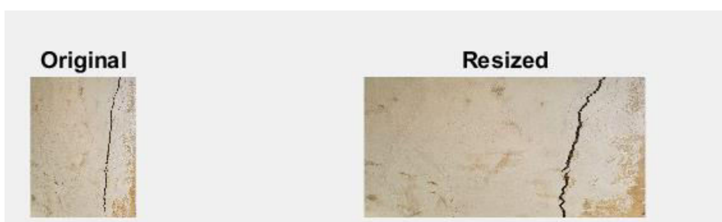


FIGURE 6.3: BASIC TRANSFORMATION APPLIED

Since the acquired images are in the form RGB; therefore, in order to apply further transformations, the images need to be converted to Gray Scale Image.

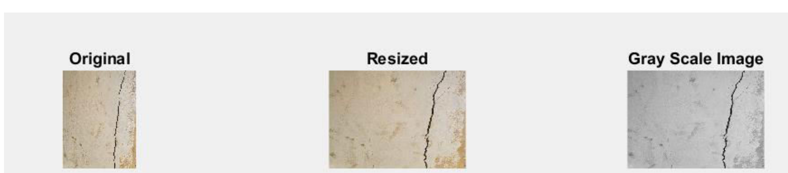


FIGURE 6.4: RGB TO GRAY SCALE CONVERSION APPLIED

Furthermore, the contrast of gray scale is equalized through white add-ons and black add-ons respectively. Moreover, the binarization is applied to the image in order to improve the quality of the image.

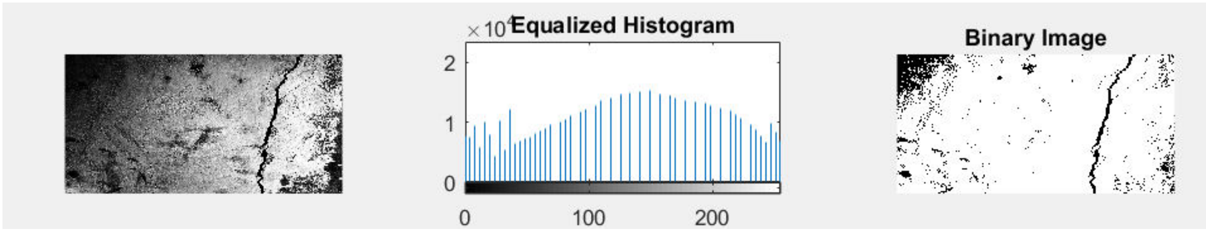
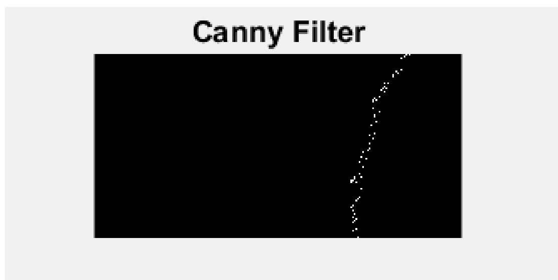


FIGURE 6.5: BINARIZATION APPLIED TO EQUALIZED IMAGE

The filters for the edge detection play a vital role in detecting the images efficiently before the process of the segmentation. The algorithms of edge detection which are as follows such as Sobel, Prewitt and Canny are applied for the better evaluation of the result. The application of filters on the image occurs through the technique of Convolutional.



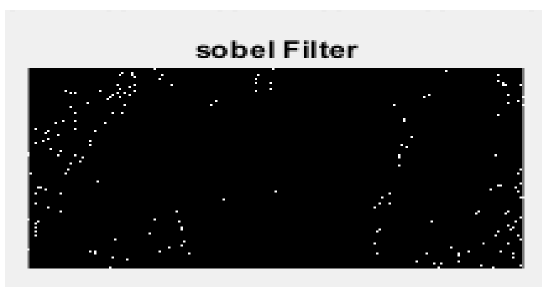
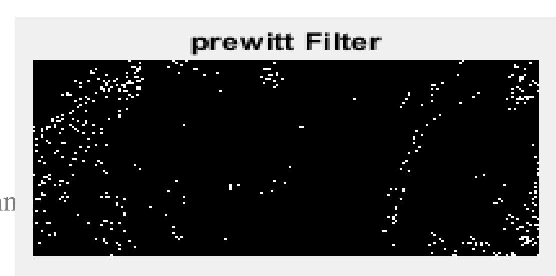


FIGURE 6.6: RESULTS OF CANNY FILTER

FIGURE 6.7: RESULTS OF SOBEL FILTER
FIGURE 6.8: RESULTS OF PREWITT FILTER



After the successful edge detection of the crack images, the process of segmentation is applied to the image that helps in locating the images and their associated boundaries within the original image. For example, the results in Figure 6.7 shows the clear white outer line around the crack. This leads to the perimeter calculation for determining the severity of the cracks.

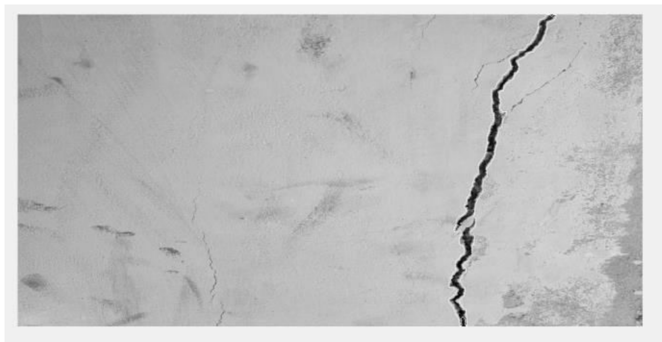


FIGURE 6.9: APPLICATION OF SEGMENTATION

In the proposed model, the perimeter being calculated to define the severity of the cracks are Mean and Ratio. The mean width is calculated by calculating the Euclidean distances for considering widths, whereas, the ratio defines the ratio of non-zero elements to that of complete number of pixels.

6.4 Comparison with other algorithms

Most of the algorithms proposed in different research papers are focused upon deep learning, supervised deep CNN, Visual Geometry Group Network and Otsu method for the detection as well as identification of the cracks. However, the research conducted by [5] was focused on evaluating the results proposed by different filters. The strategy of the research is similar to the proposed strategy where different filtering techniques are evaluated to select the most efficient technique. The Table 6.1 below shows the complete summary of different algorithms suggested at different times.

TABLE 6.1: SUMMARY OF OTHER ALGORITHMS

Research paper Name	**Author**	**Implementation Details**
Detection Of Surface Cracks In Concrete Structures Using Deep Learning	Dwevedi	~ Deep Learning: Convolution Neural Network (CNN) ~ Image Augmentations ~ Model made on Pytorch
Assessment of cracks on concrete bridges using image processing supported by laser scanning survey	Valença et.al	~MCrack-TLS (Image processing + terrestrial Laser Scanning)
Research on Crack Detection Algorithm of the Concrete Bridge Based on Image Processing	Wang et.al	~Gray-Scaling ~primary adaptive filtering multiple times ~adjusted window size ~Sobel edge detection with Otsu
Robust image-based crack detection in concrete structure using multi-scale enhancement and visual features	Liu et.al.	~ multi-scale morphological enhancement
Image analysis method for crack distribution and width estimation for reinforced concrete structures	Yang et.al	~ method based on the sub-pixel and optical flow for getting accurate width of crack
Concrete Cracks Detection Using Convolutional Neural Network Based on Transfer Learning	Su & Wang	~EfficientNetB0 using the transfer learning method ~Pre-trained weights on ImageNet ~The Adam optimizer is being deployed for updating of the network parameters ~Image Augmentation using built-in ImageDataGenerator ~Swish activation function
Road crack detection using deep convolutional neural network	Zhang et.al,	~supervised deep CNN ~ConvNets model trained on square image patches
Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone	Maeda et al	~convolutional neural networks ~The technique of end-to-end object detection is used for model training
An Intelligent Classification Model for Surface Defects on Cement Concrete Bridges	Zhu & Song	~used improved classical CNN i.e. VGG -16 (Visual geometry group network) ~rectified linear unit used as activation function ~Softmax classification layer used
Crack Detection on Concrete Surfaces Using V-shaped Features	Sato et al.	~This made use of the V-shaped detector ~Gray scaling and noise removal ~Edge Detection that calculates differences in values of pixels ~noise processing through using line definitions
Crack detection using image processing: A critical review and analysis	Mohan & Poobal	~assessment of the different available algorithms
Benchmarking Image Processing Algorithms for Unmanned Aerial System-Assisted Crack Detection in Concrete Structures	Dorafshan et al.	~For edge detection, 6 filters was used namely Laplacian of Gaussian, Roberts, Sobel, Gaussian, Prewitt and Butterworth ~The Laplacian of Gaussian from spatial was found to be most effective for real-time crack detection involving unmanned Aerial System

6.5 Algorithm Complexity

The Time Complexity of the proposed algorithm is O (n*m) since the image processing involved in this works on the matrix form of image; therefore, it considers it pixel by pixel.

The Space Complexity of the algorithm is O (n) which depends upon the number of images.

7. Implementation

7.1 Introduction

The implementation of the core crack detection was conducted using MATLAB software. The crack detection algorithm as well as the GUI of the project was made using MATLAB. The following chapter renders the complete details related to the crack detection algorithm. Moreover, it also explains the elements integrated in the GUI with consideration to importance of HCI. It also provide further details of implementation using UML diagrams i.e. Component, State Transition, Deployment, and Operational.

7.1.1 Proof of Concept:

The allotted POC of our project was to collect dataset in form of images of different concrete buildings. This dataset (images) has to be captured from the drone camera or portable camera which is going to be used in project. A total of 250 cracked and non-cracked images were gathered to serve the purpose.

7.1.2 Dataset:

Some of the images gathered for the dataset are shown below:

FIGURE 7.1: CRACK DETECTION DATASET (IMAGE 1)

FIGURE 7.2: CRACK DETECTION DATASET (IMAGE 2)

FIGURE 7.3: CRACK DETECTION DATASET (IMAGE 3)

FIGURE 7.4: CRACK DETECTION DATASET (IMAGE 4)

FIGURE 7.5: CRACK DETECTION DATASET (IMAGE 5)

7.2 Code of Crack Detection:

```
% First, the image will be uploaded on the desktop app from file system.

[filename,filepath] = uigetfile('*.jpeg');
if isequal(filename,0)
   disp('Cancel was selected');
else
   disp(['The following image was selected', fullfile(filepath,filename)]);
end
```

```
myImage = imread(fullfile(path,file));
axes(handles.axes1);
imshow(myImage);
title('Original Image','Color','white');
%subplot(4,4,1);
%imshow(image);
%title('Original');
```

% Then resize the image to a certain parameters. Code given below for resize process of image.

```
resize = imresize(myImage,[500 700]);
%subplot(4,4,2);
%imshow(resize);
%title('Resized');
```

% Next step of our algorithm is to convert the image into Black and white.

```
BW = rgb2gray(resize);
%subplot(4,4,3);
%imshow(BW);
%title('Black and White');
```

% After that, generate the histogram of the black and white image.

```
%subplot(4,4,4);
%imhist(BW);
%title('Histogram');
```

% Next step in the process is to equalize the image by observing its histogram.

```
new = histeq(BW);
%subplot(4,4,5);
%imshow(new);
%subplot(4,4,6);
%imhist(new,64);
%title('Equalized Histogram');
```

% Convert the equalized image into binary image.

```
binary = im2bw(new,0.07);
%subplot(4,4,7);
%imshow(binary);
%title('Binary Image');
```

% Apply the canny filter on the binary image.

```
BW2 = edge(binary,'canny',0.99);
%subplot(4,4,8);
%imshow(BW2);
%title('Canny Filter');
```

```
% Invert the image obtained after applying canny filter.
%Invert Image
%invert=imcomplement(BW2);
%subplot(4,4,9);
%imshow(invert);
%title('Invert');

% Image Segmentation (The Dilated Gradient Image is produced)
im0 = strel('square',9);
dilated = imdilate(BW2,im0);
%subplot(4,4,10);
%imshow(dilated)
%title('Dilated Gradient Masked Image')

% Binary Image with Filled Holes
filled = imfill(dilated,'holes');
%subplot(4,4,11);
%imshow(filled)
%title('Binary Image having filled holes')

im1 = strel('disk',1,4);
finalimage = imerode(filled,im1);
finalimage = imerode(finalimage,im1);
%subplot(4,4,12);
%imshow(finalimage)
%title('Segmented Image');

% One of the important step is to create an outline on the Image.
% The outlined image is basically original image with white outline
outlined = bwperim(finalimage);
segmentout = BW;
segmentout (outlined) = 255;
%subplot(4,4,13);
%imshow(segmentout)
%title('Outlined Original Image')
axes(handles.axes3);
imshow(segmentout);
title('Outlined Original Image','Color','white');

% The Final Image will be converted to binary Image
masked = imbinarize(double(finalimage));

% The holes in the masked image above will be filled
masked = imfill(masked, 'holes');
```

```
% The largest blob is taken only
masked = bwareafilt(masked, 1);

% The skeleton is computed
skeletonimage = bwmorph(masked,'skel',Inf);
%title('Thinned')

% Euclidean distance of the image is computed
image3 = bwdist(~masked);
%title('Distance Transform Image');
diameterImage = 2 * image3.* single(skeletonimage);
%title('Diameter Image');

% The widths are extracted where the diameter is not equal to zero
widths = diameterImage(diameterImage > 0);

% The histogram with widths is shown in terms of pixels
%subplot(4, 4, 16);
%histogram(widths);
%grid on;
%xlabel('Width in Pixels');
%ylabel('Count');

% The mean width of all widths is calculated (1st Parameter)
meanwidth = mean(widths)

% A Red color line is shown indicating mean width
%line([meanwidth, meanwidth], ylim, 'LineWidth', 2, 'Color', 'r');
%caption = sprintf('Histogram of Widths.  Mean Width = %.1f Pixels', meanwidth);
%title(caption);

%Calculation of Ratio (2nd parameter)
ratio = nnz(masked)/numel(masked);
set(handles.text13,'String',meanwidth);
set(handles.text9,'String',ratio);

%Convert ratio from pixels to mm
ratioInmilli = ratio*0.26458333333333;

%Convert mean from pixels to mm
meanwidthInmilli=meanwidth*0.26458333333333;

if((meanwidth>=1 && meanwidth<=5)&&(ratio>=0.001 && ratio<=0.01))
   analysis='Slight';

elseif((meanwidth>5 && meanwidth<=15)&&(ratio>0.013 && ratio<=0.05))
```

```
    analysis='Moderate';

elseif((meanwidth>15 && meanwidth<=25)&&(ratio>0.05 && ratio<=0.1))
    analysis='Severe';

else
    analysis='Negligible';
end
```

7.3 Project Overview:

Our project mainly consist of mainly two parts software application and hardware implementation. Software application mainly consist of algorithm and interactive screens. Our algorithm is divided into many steps which are given below:

- Upload data.
- Resize images.
- Convert data into Gray Scale
- Equalize the images by observing its histogram.
- Convert it into Binary image and apply canny filter.
- Apply image segmentation.
- Calculate the Mean width (1^{st} Decision Parameter)
- Calculate the ratio of the image. (2^{nd} Decision Parameter)
- Provide the analysis of crack severity on the basis of 2 parameters

Second part of our software application is interactive screens. Project application is designed in such a way that it will assist the user in performing their tasks. The hardware implementation of project mainly consist of drone. Drone with particular specification is used to capture the images of concrete structure cracks.

7.4 Project Objectives:

This project will meet the following objectives:

- The project has the major objective to assist the inspection team.
- Drone will assist in capturing the images of cracks existing on high-rise buildings.
- Inspector can easily observe the crack in concrete structure through screen using direct the drone.
- In Desktop App, the crack detection on the images will take place.
- Application screens will provide the best experience to the inspector.
- The markers could be used for tracking these cracks. Since, there are different types of cracks boundaries are shown below:
  - Negligible (Less than a mm)
  - Slight (1 to 5 mm)
  - Moderate (5 to 15 mm)
  - Severe (Up to 25 mm wide)
- This type of crack detection technique will be cost effective as well as time-saving since the cracks will be measured faster and efficiently.

The out-of-scope objectives for the project are covered in Appendix B.

7.5 Technical Interface:

Technical interface consists of the HCI representation and modes of cognitive processes:

- Attention
- Reading, speaking and listening
- Memory
- Perception and recognition
- Learning
- Problem-solving, planning, and decision-making

7.5.1 Attention:

The attention of our project consists of the clear vision to an inspector to solve his query after a part inspector also show a warning message if drone did not work etc. However, with regard to the attention interface, attention and attractive interface gives more concentration and best work to every user or inspector. This aspect also involves making those salient features evident which needs to stand out. This approach is implemented through elements such as colors, sequencing, animation, visual aspects, images, and ordering. However, it should be ensure that the interface is not cluttered with a lot of information or visual elements to avoid distraction for the user. Consider Fig. 5.11, 5.12 and 5.13 for the best attention implementation in the current project.

7.5.2 Perception and Recognition:

Perception considers being a deep following of standards such that the GUI looks standardized as compare to worldwide projects because the world considers globally the format of text profiles, icons, etc. So the clear perception and recognition helps to promote the project and fine it is clear all prospective.

7.5.3 Memory:

Our memory feature considers processing crack images and find out the damage category for an inspector and the inspector found easily the damage or the crack is severe or ignorable. Also, calculate all parameters for which the algorithm is already designed in the memory of the project. Through the GUI, it was ensured that the memory of the user is not overloaded with a lot of tasks to be carried out; therefore, it was aimed to be kept simple and clear. The interface design should focus on making the users recognize rather than recall. Such elements need to be integrated in the GUI which helps them in remembering the interface as shown in Figure 5.5.

7.5.4 Learning:

A user manual to be considered for our user how the system works how pictures upload how to detect crack and overall operations like operate drone be the part of learning for our users. To the point of my team, we considered who is our target audience or user so we didn’t make complicated to our user according to their knowledge.

7.5.5 Reading, speaking, and listening:

The reading, speaking and listening factor consideration allows the ability to comprehend the fact that all people are different in terms of the reading, listening or speaking ability. Some people give preference to listening as compared to reading, Moreover, reading is quicker in comparison with the other two. However, less cognitive efforts are required in the listening process. The speech recognition systems are an efficient way to help such users having difficulty in either typing or suffer through lack of time to type. One such example can be Google Voice app where the users can search through using their speaking ability. So, in the future, another team can work with voice reorganization add on our project for more easiness to our inspector.

7.5.6 Problem-solving, planning, reasoning, and decision-making:

During designing of the screens, the elements involving reflective cognition were also taken into account. For example, thinking about the actions, options, and consequences from the user’s perspective. Another important question to consider was that is our project is useful for our target user? Yes of course it is useful. Is our project is problem-solving? Yes, it is a problem-solving project because before this our target user does his job manually and it’s too risky for his life and time-consuming also so absolutely it is problem-solving.

7.6 Diagrams

7.6.1 Component Diagram:

The component diagram of the project is shown in Figure 7.6 above. This component diagram describes the functionalities of the system. The sign in / sign up database component contains the inspector details that is ID, password. When inspectors sign in or sign up in the system, all inspector details will save in the inspector database.

The capture of the image component consists of three sub-components which are pre-processing, define parameters, and define severity, and then the result in the component is used to generate a final result of crack’s severity.

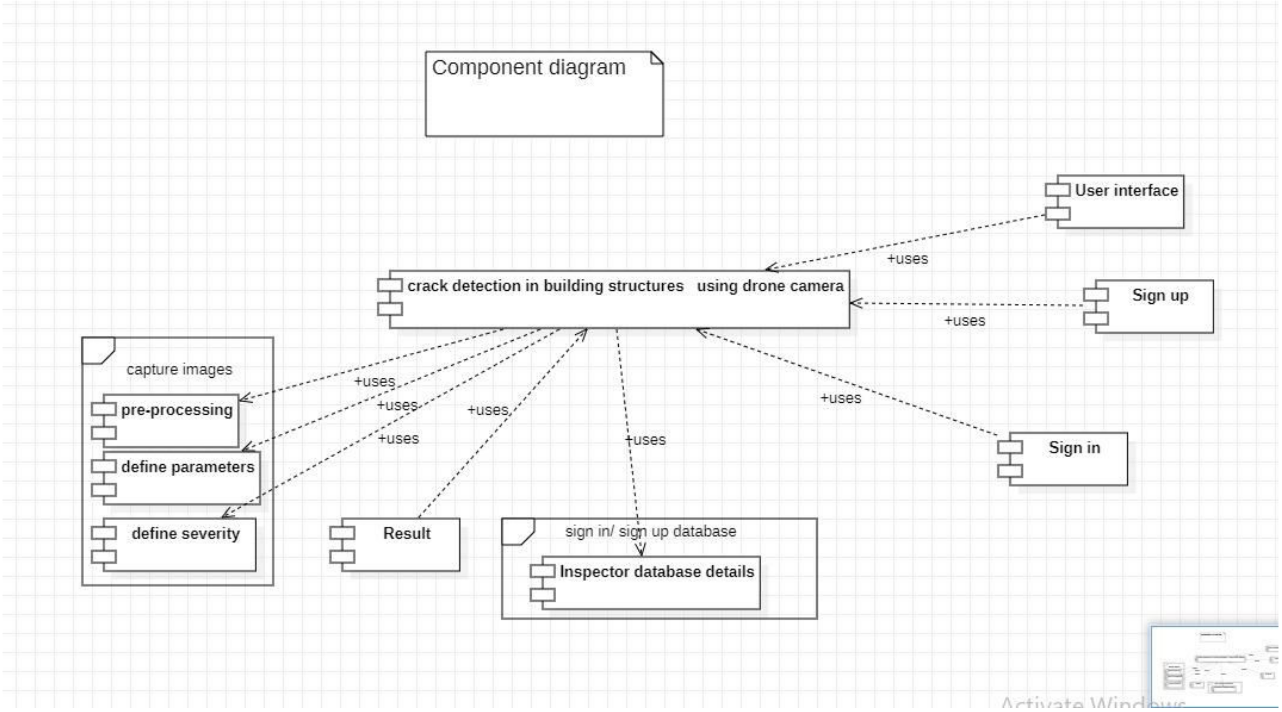


FIGURE 7.6: COMPONENT DIAGRAM

7.6.2 Operational Diagram

The Figure 7.7 shows the operational architecture of the system. The inspector will sign in / sign up for the system. The inspector will capture the image through a drone and monitor the building through his drone application on mobile. The captured images will save on the desktop for further processing, and the final result will appear to the inspector on the desktop screen.

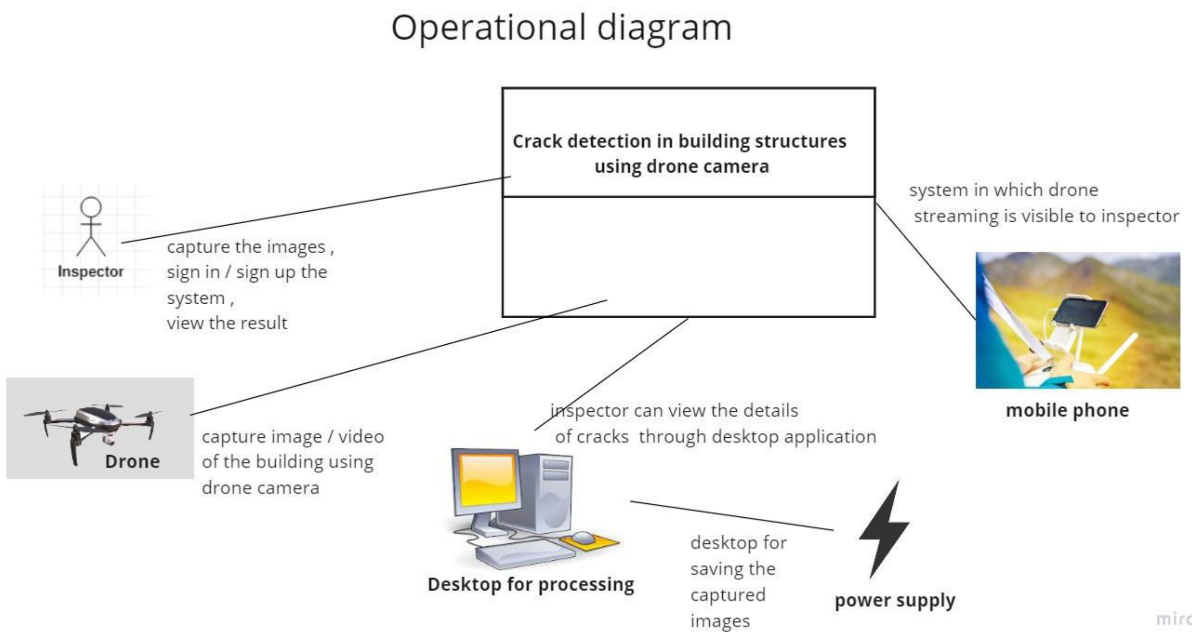


FIGURE 7.7: OPERATIONAL DIAGRAM

7.6.3 Deployment diagram

The Figure 7.8 shows the deployment diagram in which consists of three nodes that are

1. Inspector
2. Database
3. Application

- The Inspector node consists of calculating the severity of cracks and as well as consist of capture image.
- The database is shown in the database node in which inspector sign in / sign up details will save.
- The application node consists of two components:-
  1. Login .m
  2. View results

In login, the node-inspector will sign in or sign up the system and will perform further functionality, will get the final result, and view the results of cracks.

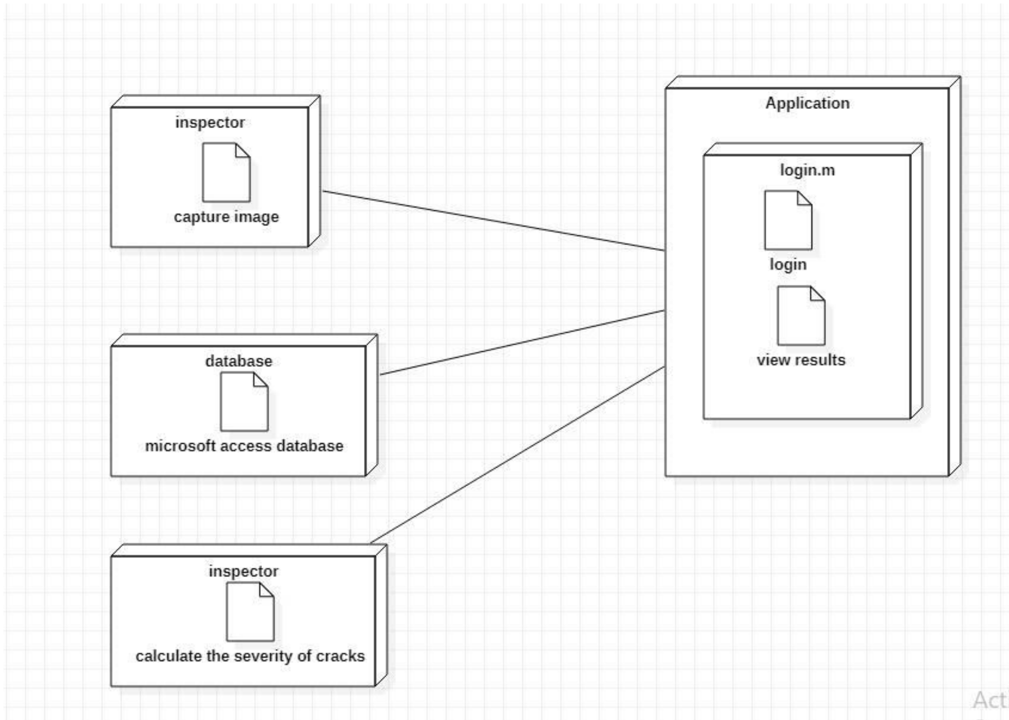


FIGURE 7.8: DEPLOYMENT DIAGRAM

7.6.4 State transition Diagram:

The state transition diagram shows all possible transition states of the system. The state transition diagram is shown in the Figure 7.9. Firstly inspector will capture the image through a drone and then save it in a system. After that he logs in the system and if he has an account then he will sign in to the system otherwise sign up for the system. Dashboard will appears on the desktop in which the inspector has the choice to select the module. There are three modules in our application crack detection, inspector profile update option, and logout. In the crack detection module, the inspector selects the image and then apply further processing and give the final result of the ratio and mean of cracks images. The severity of the cracks will be shown on the basis of these parameters.

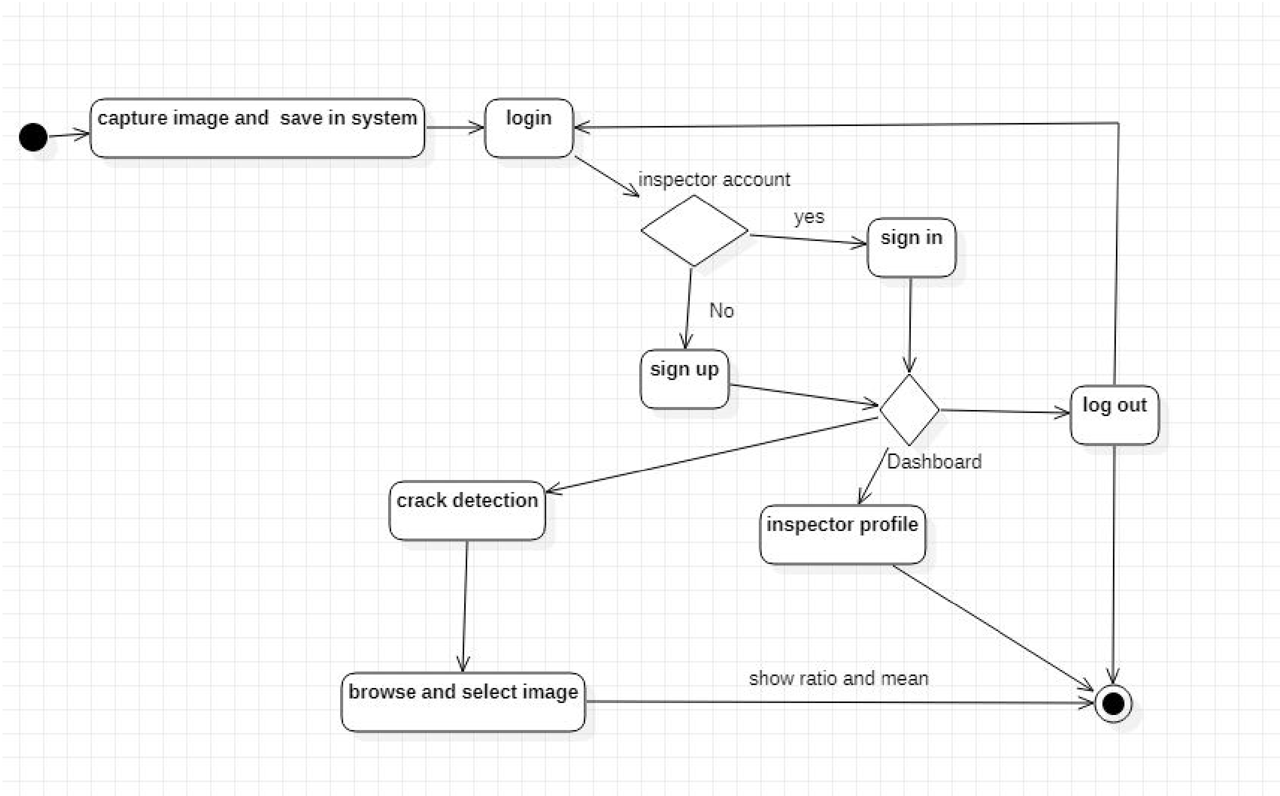


FIGURE 7.9: STATE TRANSITION DIAGRAM

7.7 Deployment Architecture with Strategy

7.7.1 Deployment

In development of this project, the software services for end users (inspectors) would be reliable enough that he can work without any hurdle. As tested our product on several data and it produces the desired result. Project team would also create a resource through which management can get the feedback of our users. Feedback plays an important role in the future development of product.

Project team will surely communicate with the end users to before launching the product. They would remind them that this is our new product and this product needs some fixes.

For a successful deployment of our product, the required resources are given below:

- If talk about hardware requirements, our product requires a compatible drone which would be able to collect the data of concrete structure cracks. Another hardware requirement for this product would be a compatible desktop computer or Laptop. This device runs product software.

- Consider the software requirement for product. The software requirement of this product consist of MATLAB program. MATLAB programs contains the algorithm which helps in identifying the crack in data.

7.7.2 Deployment Strategy

A deployment strategy plays an important role in the future product enhancement. As it is considered as an approach to change or upgrade the product. The deployment strategy that is being used for this product will be blue-green deployment. In blue-green deployment, the blue version (new version) of the product is developed and then prepared for testing and evaluation. While the end users (inspectors) still uses the green version (stable version) of product. When the blue version (new version) is ready, the users are switched to the new version. Using this strategy, if any bug or problem appears in the blue version (new version), it can switch our product to the green version.

8. Testing

8.1 Introduction

This chapter pertains to describe the testing phase of the crack detection algorithm and application as a whole system. The unit test cases were developed for each module of the system with their respective objective. Moreover, these unit test cases will be evaluated on the basis of its expected result and the output. The chapter is divided into conducting the 2 testing types i.e. Black Box and White Box Testing. In Black Box Testing is being conducted without any prior knowledge about the internal functionalities of the system, whereas, the white box involves the evaluation of internal structure and algorithms of the system. The tester in the case of black box assumes the system to be a black box which takes input and produces output [17].

8.2 Black Box Testing

The tables below shows the black box testing being conducted to test the functionalities of the system irrespective of internal structure. The test case, its expected result and the output produced are considered to evaluate whether the system passes or fails.

8.2.1 Unit Test: Crack Detection

Objective: Successful detection and showing of the outlined original image showing crack

TABLE 8.1: CRACK DETECTION (BLACK BOX TESTING)

S.no	Test Case	Expected Result	Output	Pass/Fail
1	The user clicks on the Crack Detection push button on dashboard	The dashboard screen of Add Crack is prompted	The “Add Crack” screen is prompted	Pass
2	The user clicks on the “Browse & Detect” push button	The prompt to select file from local drive is prompted	The screen to select files is shown	Pass
3	The user selects Browse & Detect	The files with extension “.jpg” are only shown	The images with .jpg extension is shown	Pass

4	The user selects the image from the local drive	The image is selected	The image is selected	Pass
5	The user clicks on "Open"	The original and outlined images are prompted on the axes along with mean and ratio	Both the original and outlined images are prompted on the axes along with mean and ratio	Pass
6	The user click on "Cancel" button	The user is directed back to Add Crack screen	The Add Crack screen is shown	Pass

8.2.2 Unit Test: Sign In

Objective: The user logs in the application and whether the dashboard is displayed or not

TABLE 8.2: SIGN IN (BLACK BOX TESTING)

S.no	Test Case	Expected Result	Output	Pass/Fail
1	The user enters correct username	The user is being displayed on screen of login interface	The username is visible on the interface	Pass
2	User enters the correct password	The password is displayed in asterisk form	The password is visible in the alphanumeric form	Fail
3	The user clicks on "Sign In"	The dashboard is displayed	The dashboard is displayed with prompt "Successful Sign In"	Pass
4	The user enters incorrect username	The user is being displayed on the login interface	The username is visible on the interface	Pass
5	User enters the incorrect password	The password is displayed in asterisk form	The password is visible in the alphanumeric form	Fail
6	The user clicks on "Sign In"	The dashboard is not displayed	The prompt "Invalid ID Pass" is displayed	Pass

7	The user misses entry of information is any of the field	No successful logging In	The prompt “Invalid ID Pass” is displayed	Pass

8.2.3 Unit Test: Sign Up

Objective: The user able to sign up and directly logs in the application where the dashboard is made visible

TABLE 8.3: SIGN UP (BLACK BOX TESTING)

S.no	Test Case	Expected Result	Output	Pass/Fail
1	The user clicks on “Sign Up” from Login Screen	The Sign Up interface is displayed	The Sign Up interface is displayed	Pass
2	The user enters more than 10 characters in “Name” field	Invalid no of characters	The prompt “Not allowed more than 10 characters in Name” is displayed	Pass
3	The user enters exact 10 or less than 10 characters in “Name” field	Valid no of characters	No error is prompted	Pass
4	The user enters more than 10 characters in “Email” field	Invalid no of characters	The prompt “Not allowed more than 10 characters in Email” is displayed	Pass
5	The user enters exact 10 or less than 10 characters in “Email” field	Valid no of characters	No error is prompted	Pass
6	The characters in “Email” field have space	Invalid Email	The prompt “No spaces allowed in email” is displayed	Pass
7	The user enters characters alphabets in Inspector ID	Wrong Inspector ID should be prompted	The prompt “Missing field(s)” is displayed	Fail
8	The user more than 4/less than 4 numbers in Inspector ID	Invalid Inspector ID	The prompt “Invalid ID! Must be 4 digits” is displayed	Pass

9	The user enters less than 5 or more than 10 characters in “Password” field	Invalid Password Length	The prompt “Password length not suitable” is displayed	Pass
10	The user enters password without any digit	Invalid Password	The prompt “At least 1 upper case should exist” is displayed	Pass
11	The user enters password without any uppercase letter	Invalid Password	The prompt “At least 1 upper case should exist” is displayed	Pass
12	The user enters characters in range 5 to 10 with 1 upper case letter and digit in “Password” field	Valid Password Length	No error prompted	Pass
13	The user enters already registered inspector ID	Inspector ID already exist	The prompt “ID already exist” is displayed	Pass
14	The user misses entry of information is any of the field	No successful logging In	The prompt “Missing field(s)” is displayed	Pass
15	The user enters all required field properly and clicks on “Sign Up” button	Successful Log in the application	The prompt “Successful Signed Up” is displayed and Home dashboard is prompted	Pass

8.2.4 Unit Test: Logout

Objective: The user gets logout of the application

TABLE 8.4: LOG OUT (BLACK BOX TESTING)

S.no	Test Case	Expected Result	Output	Pass/Fail
1	The user clicks on “Logout” button	A prompt is displayed	The prompt “Are you sure you want to logout” is displayed	Pass
2	The user clicks on “Yes” on logout interface	User is directed to Login screen	The Login interface is displayed	Pass

3	The user clicks on "No" on logout interface	No change occurs	The user remains on the same interface	Pass

8.3 White Box Testing

The tables below shows the output of the white box testing technique which tests the internal functionalities of the system. The test case, its expected result and the output produced are considered to evaluate whether the system passes or fails.

8.3.1 Unit Test: Crack Detection

Objective: Successful detection and showing of the outlined original image showing crack

TABLE 8.5: CRACK DETECTION (WHITE BOX TESTING)

S.no	Test Case	Expected Result	Output	Pass/Fail
1	The original image undergoes basic transformation	The original image gets resized as per parameters	The original image gets resized	Pass
2	The resized image undergoes Gray Scale Conversion	The original image changes from RGB to Gray Scale	The original image changes from RGB to Gray Scale	Pass
3	The Histogram of Gray Scale is made	The Histogram is produced	The Histogram is produced	Pass
4	The equalized image changes to binary	The binary converted image is shown	The binary converted image is shown	Pass
5	The Filter is applied	The image with clear edge is shown	The image with clear edge is shown	Pass
6	The segmentation is applied	The original image with outlined crack is shown	The original image with outlined crack is shown	Pass

7	The masking is applied	The distance transform Image is displayed	The distance transform Image is displayed	Pass
8	The thinning is applied	The thinned Image is displayed	The thinned Image is displayed	Pass
9	Euclidean distance transform is applied on Binary Image	The distance transform Image is displayed	The distance transform Image is displayed	Pass
10	The diameter image is generated	The Diameter Image is displayed	The Diameter Image is displayed	Pass
11	The Widths and mean width are calculated	The Histogram of Widths with mean width is displayed	The Histogram of Widths with mean width in Red Line is displayed	Pass
12	The formula of ratio is applied	The ratio is calculated	The ratio is calculated	Pass

8.3.2 Unit Test: Sign In

Objective: The user logs in the application and whether the dashboard is displayed or not

TABLE 8.6: SIGN IN (WHITE BOX TESTING)

S.no	Test Case	Expected Result	Output	Pass/Fail
1	Verify if the user is able to sign in with correct user name and password	The username and password is checked against database. The user logs in the system.	The user successfully log in	Pass
2	Verify if the user is able to sign in with incorrect user name and password	The username and password is checked against database. The user is not able to log in the system.	The user is not able to successfully log in	Pass
3	Verify if password field is in form of asterisk	The password should be in form of asterisk	The password is in alphanumeric form	Fail

4	Verify if the user is able to log in on missing any of the field	The user is not able to login because of missing fields	The user is not able to successfully log in	Pass

8.3.3 Unit Test: Sign Up

Objective: The user able to sign up and directly logs in the application where the dashboard is made visible

TABLE 8.7: SIGN UP (WHITE BOX TESTING)

S.no	Test Case	Expected Result	Output	Pass/Fail
1	Verify if the user enters correct size of name field	No error is displayed	The user successfully log in on pressing “Sign In”	Pass
2	Verify if the user enters correct size of email field without spaces	No error is displayed	The user successfully log in on pressing “Sign In”	Pass
3	Verify if the user is able to sign in on missing any of the field	The user is not able to sign in because of missing fields	The user is not able to successfully sign in	Pass
4	Verify if the user is able to sign in with password having no upper case letter and digit	The user password should have at least 1 upper case letter and 1 digit.	The user is not able to sign in	Pass
5	Verify if the user is able to sign in again with same Inspector ID	The user should not able to Sign In again if the ID is available in the Database	The prompt of “Already existing ID” is displayed	Pass
6	Verify if the user is able to sign in with not equal to 4 digits	The user should have a 4 digit ID to sign in	The user is not able to sign in	Pass

8.4 Visual Representation of Testing Results

8.4.1 Black Box Testing

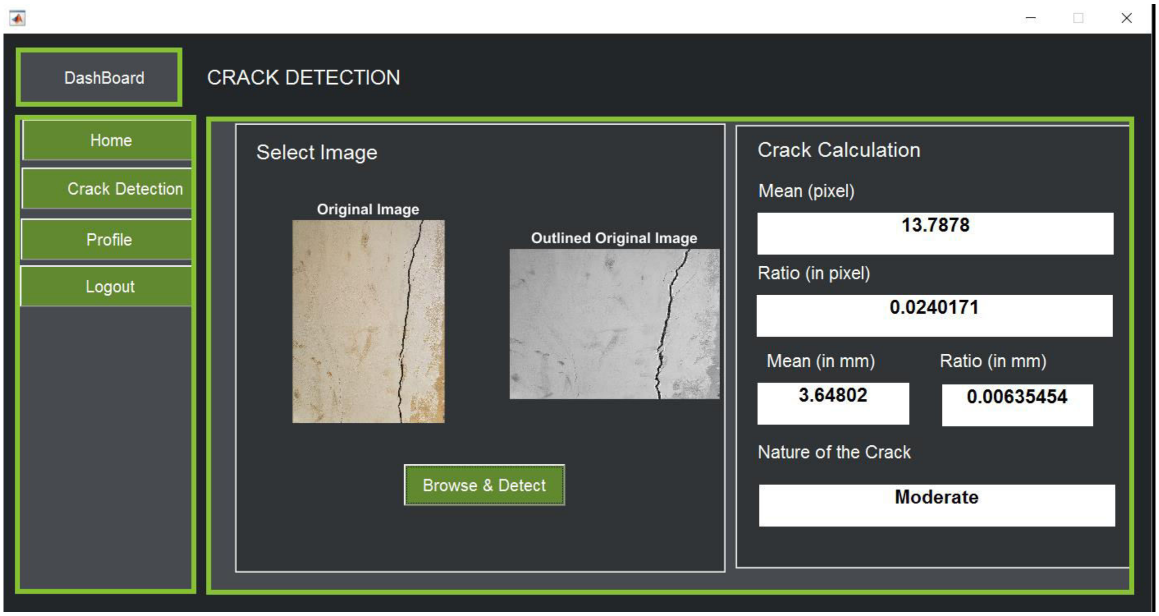


FIGURE 8.1: BLACK BOX TESTING ON GUI (IMAGE 1)

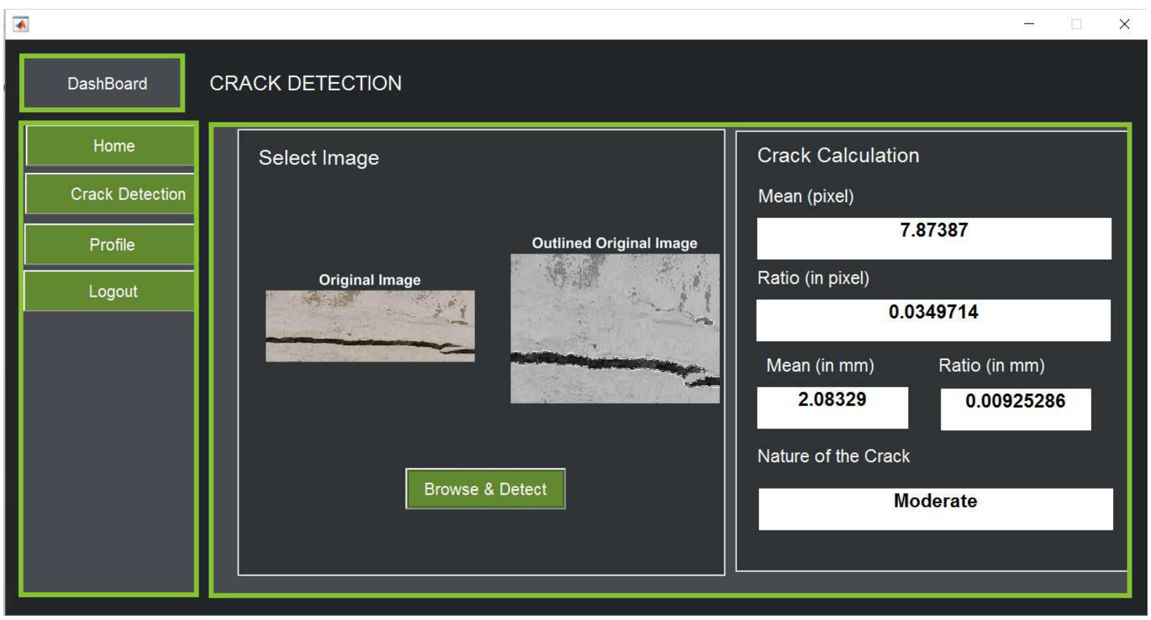


FIGURE 8.2: BLACK BOX TESTING ON GUI (IMAGE 2)

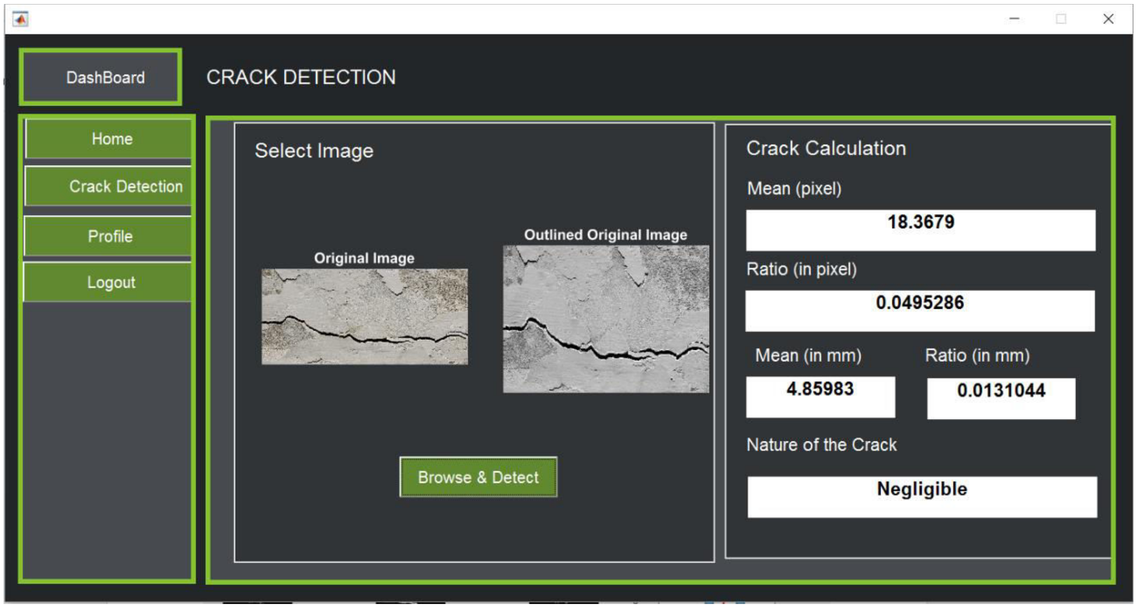


FIGURE 8.3: BLACK BOX TESTING ON GUI (IMAGE 3)

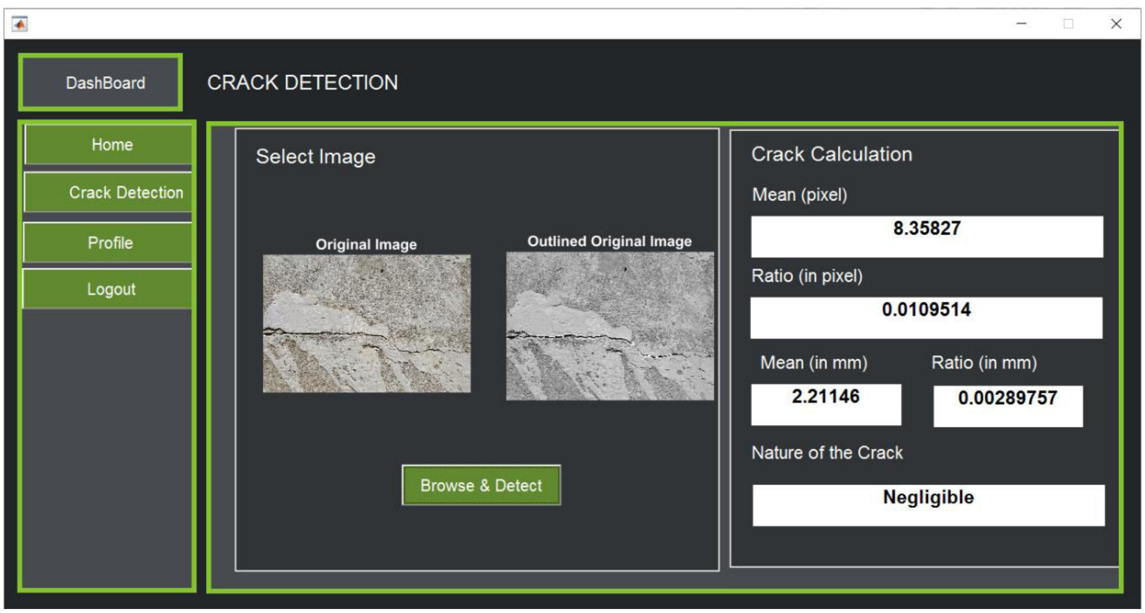


FIGURE 8.4: BLACK BOX TESTING ON GUI (IMAGE 4)

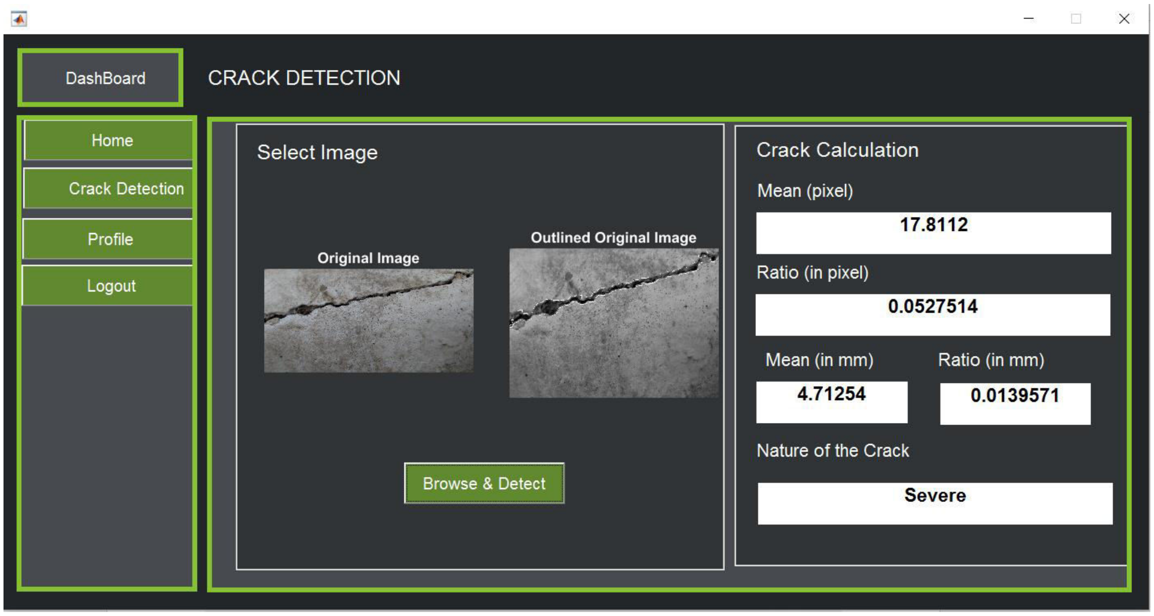


FIGURE 8.5: BLACK BOX TESTING ON GUI (IMAGE 5)

8.4.2 White Box Testing

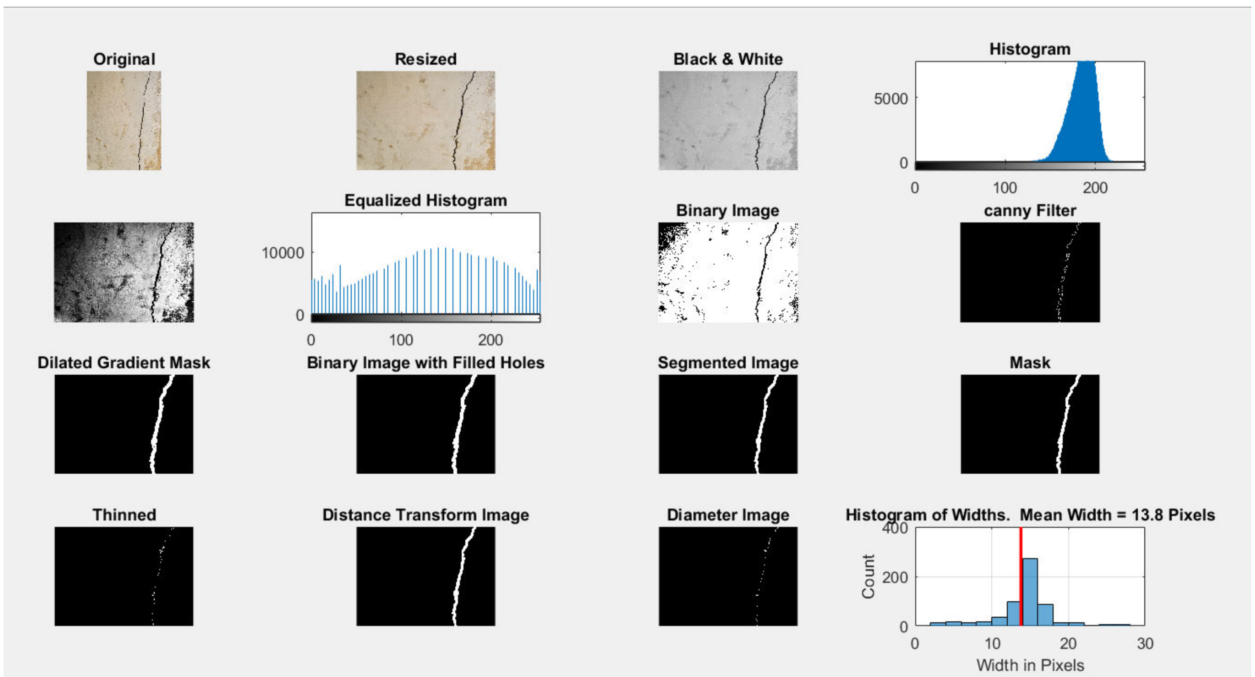


FIGURE 8.6: WHITE BOX TESTING ON GUI (IMAGE 1)

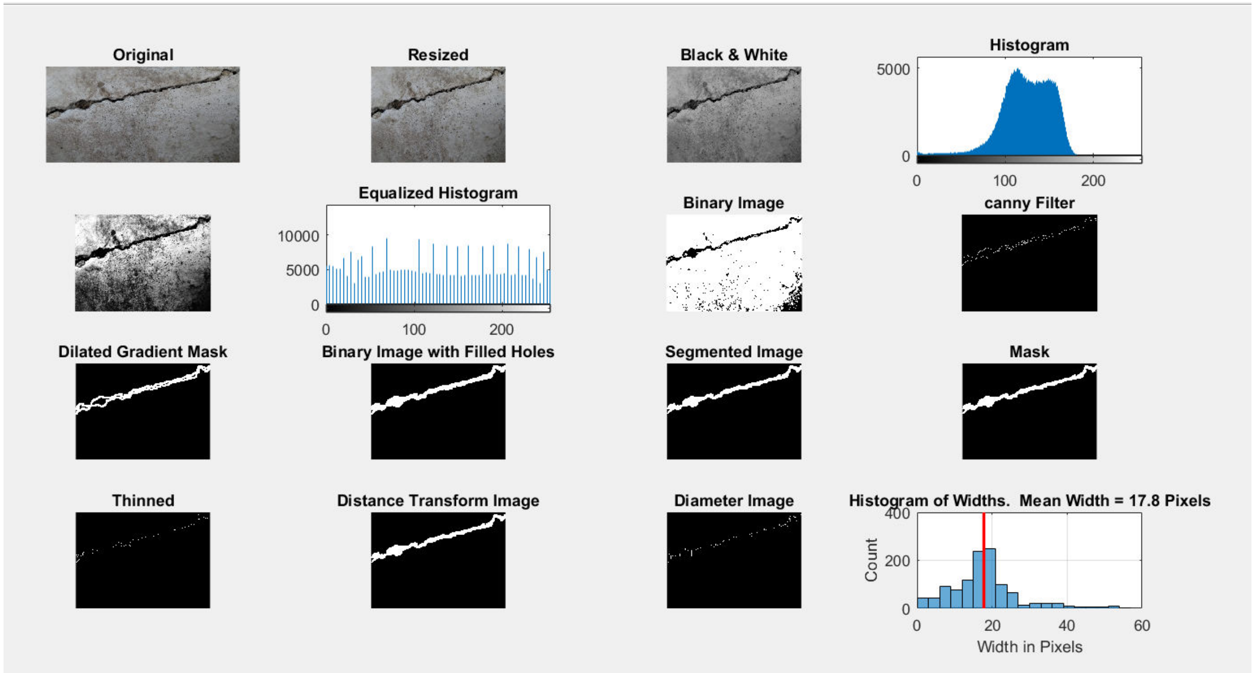


FIGURE 8.7: WHITE BOX TESTING ON GUI (IMAGE 2)

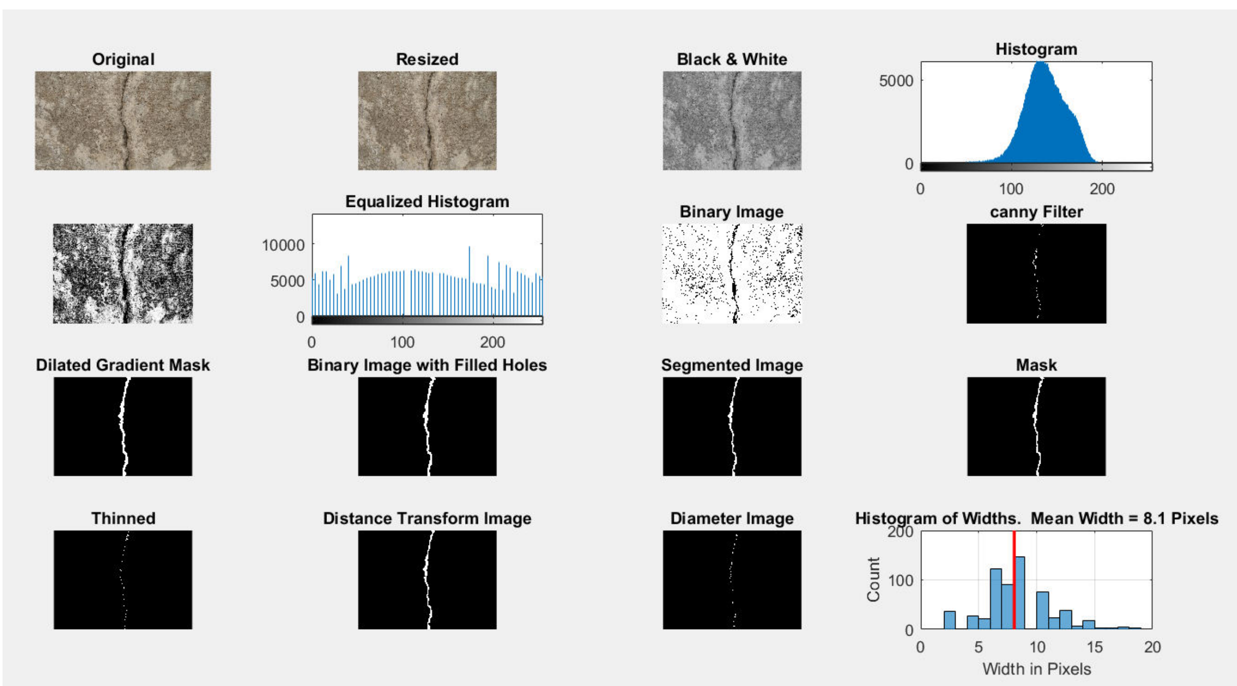


FIGURE 8.8: WHITE BOX TESTING ON GUI (IMAGE 3)

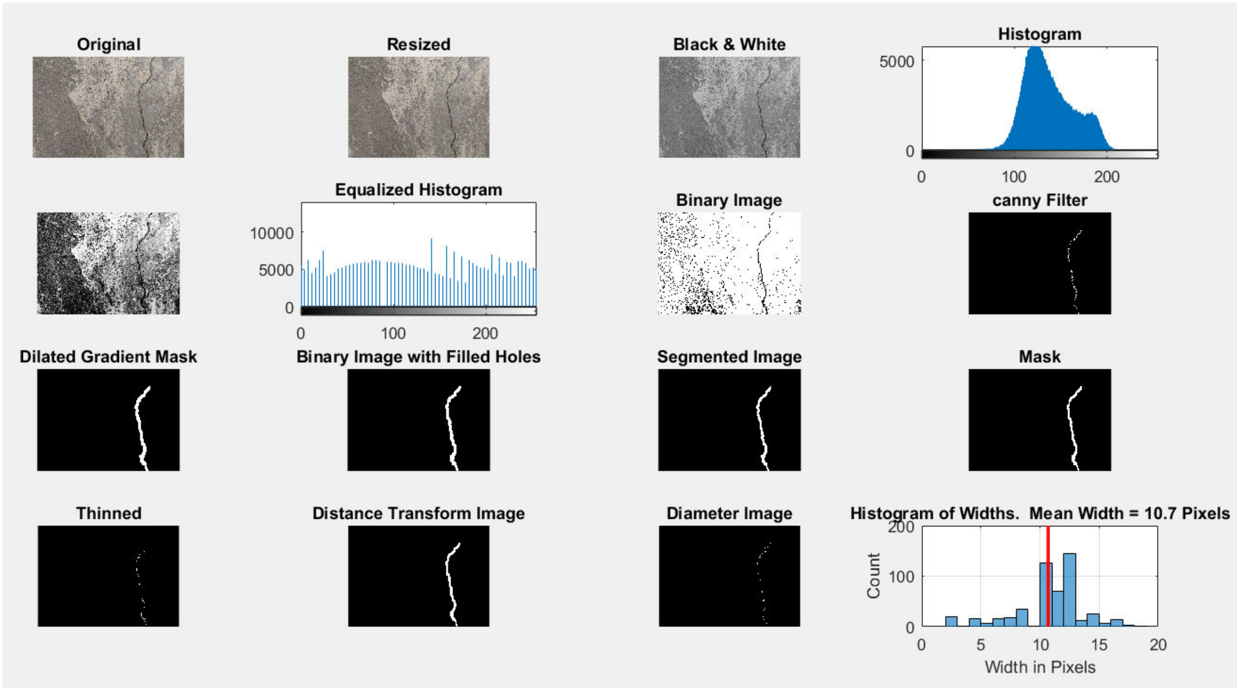


FIGURE 8.9: WHITE BOX TESTING ON GUI (IMAGE 4)

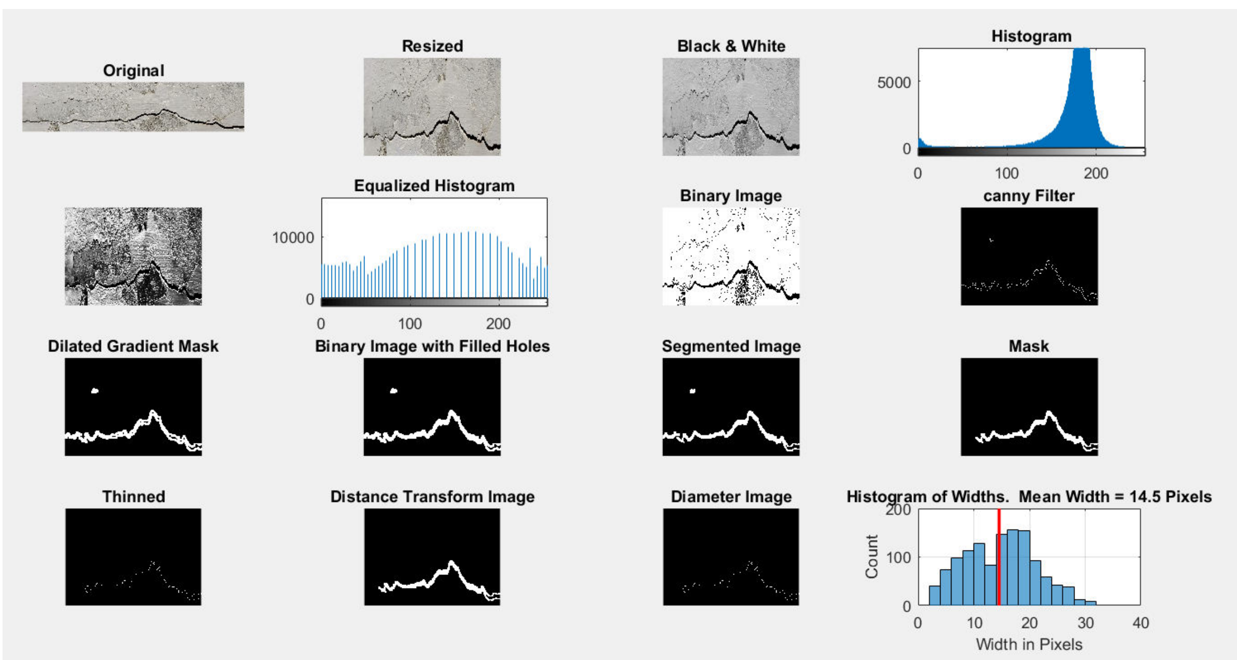
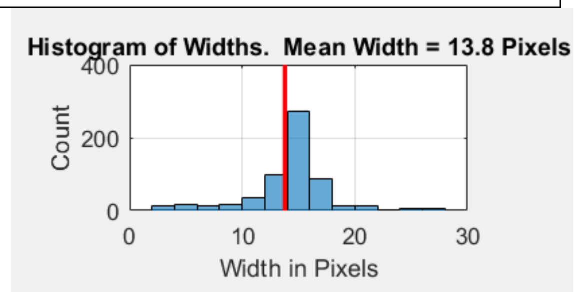
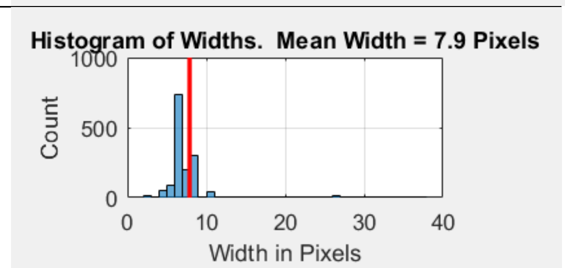
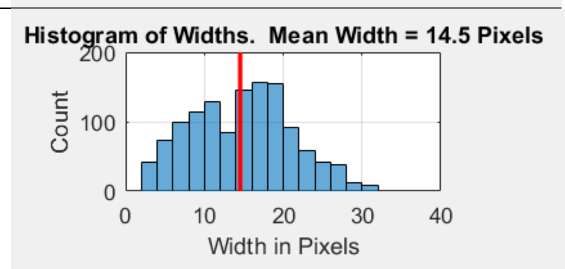
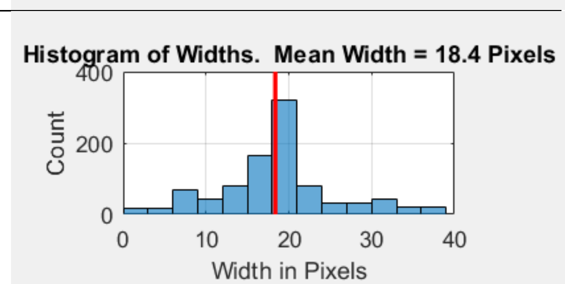


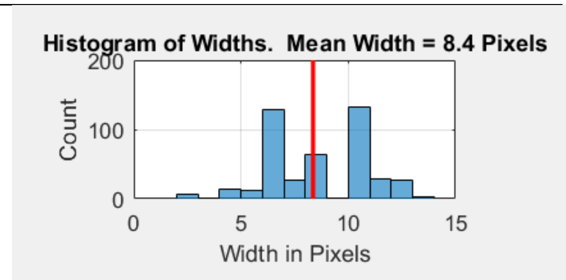
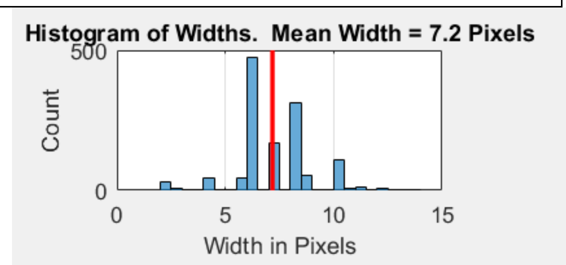
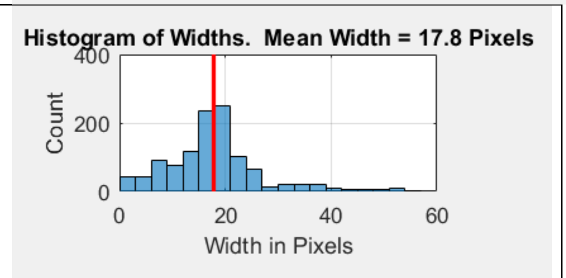
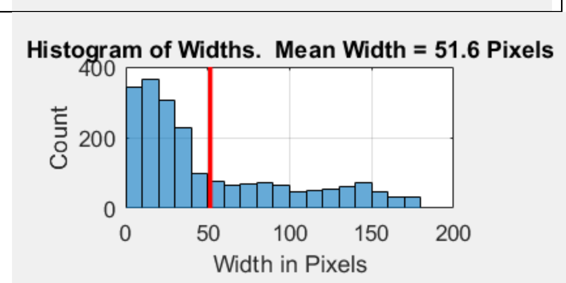
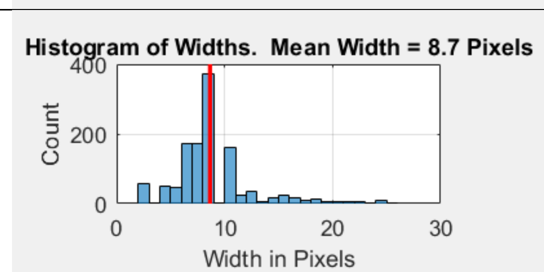
FIGURE 8.10: WHITE BOX TESTING ON GUI (IMAGE 5)

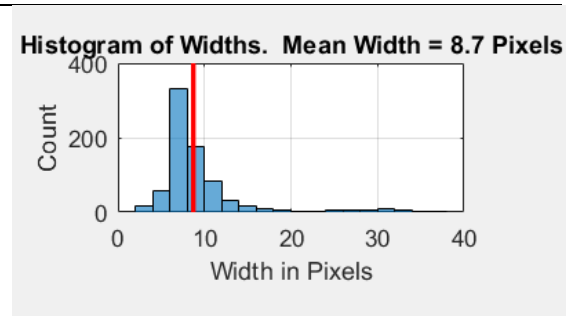
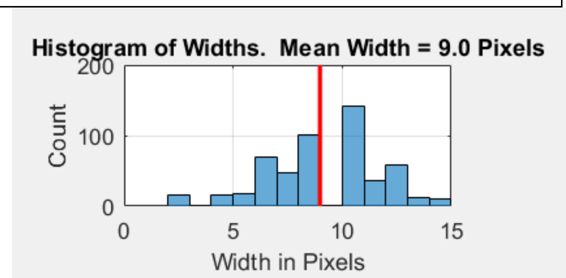
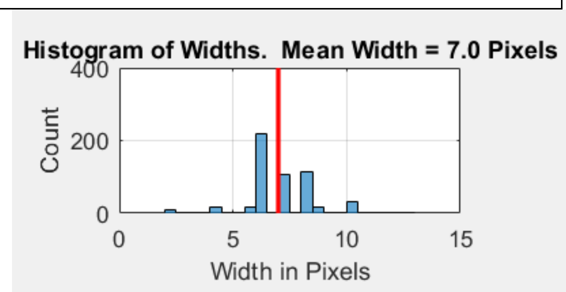
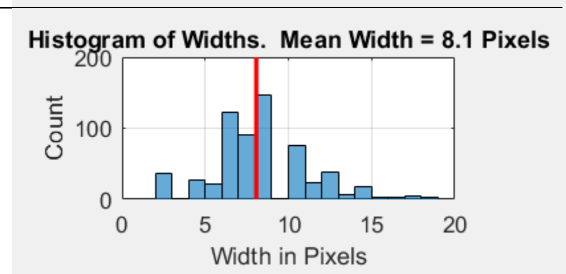
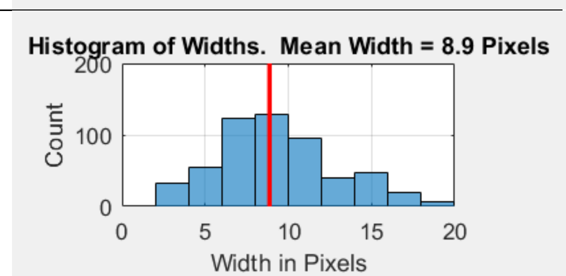
8.4.3 Summary Table of White Box Testing

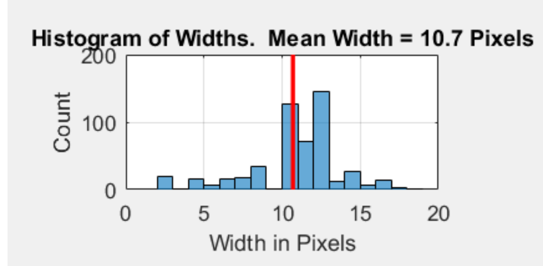
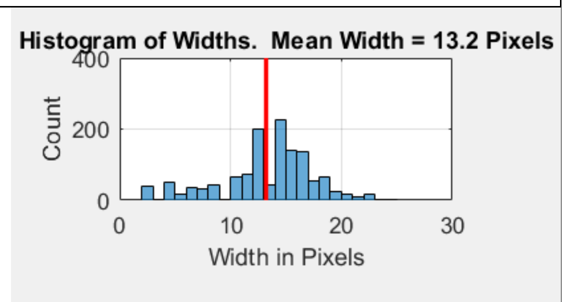
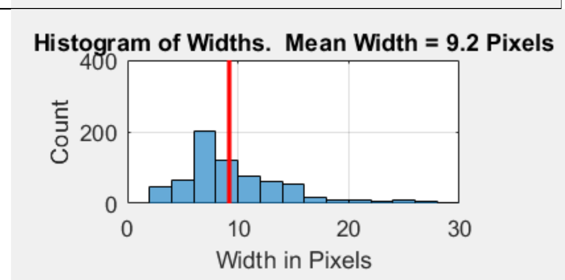
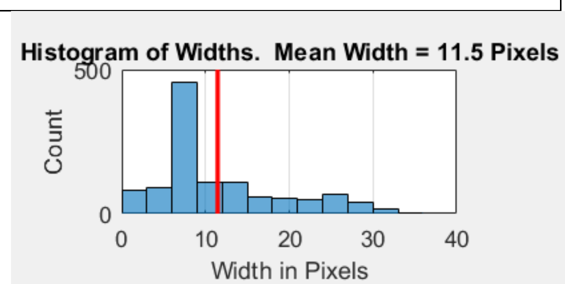
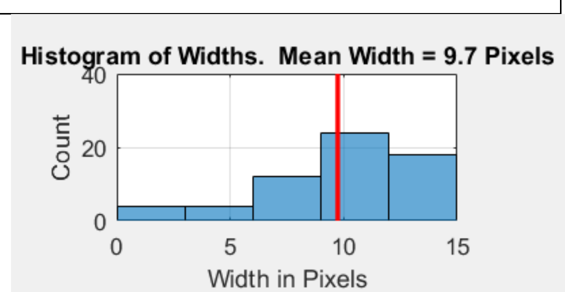
The table below presents the summary of white box testing results showing the original image along with its parameter value i.e. Ratio and Mean as well as Mean Histogram.

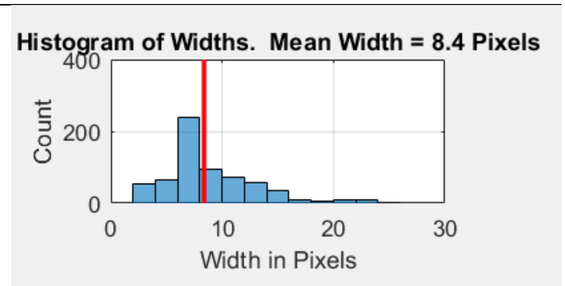
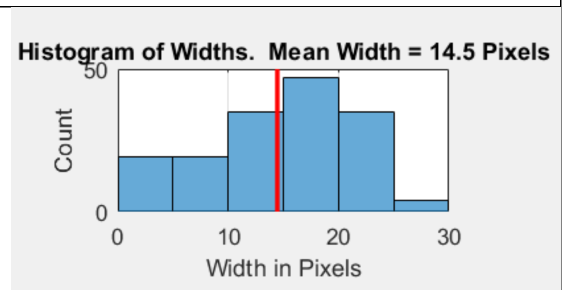
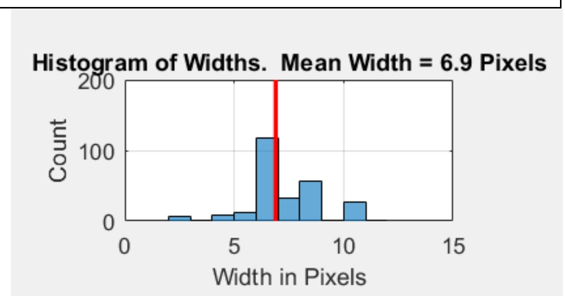
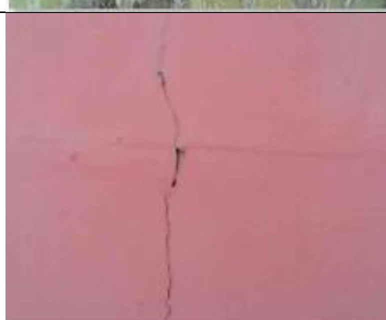
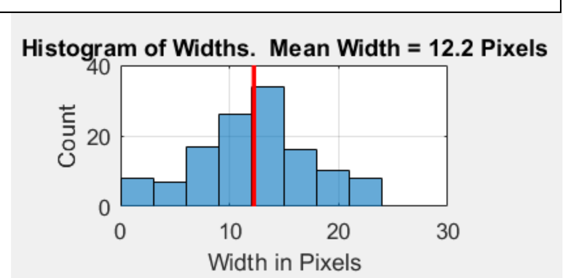
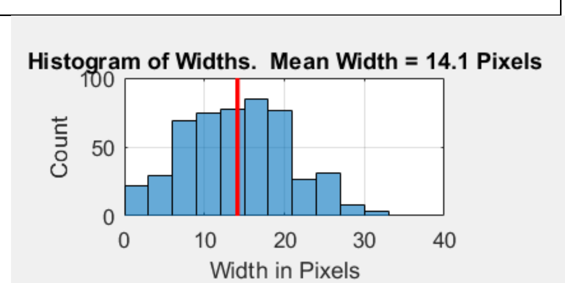
TABLE 8.8: SUMMARY OF WHITE BOX TESTING

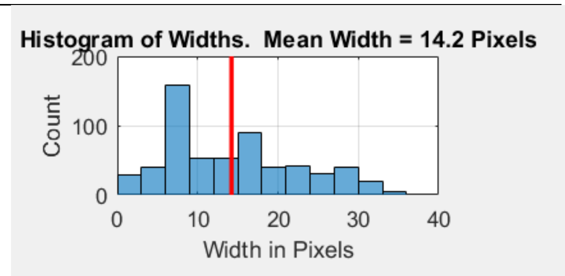
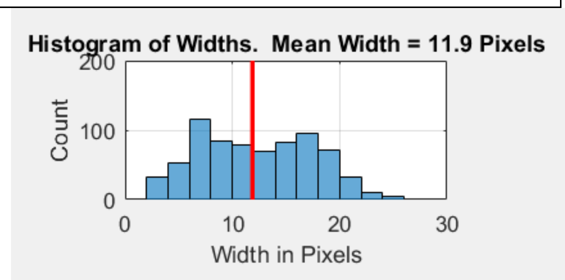
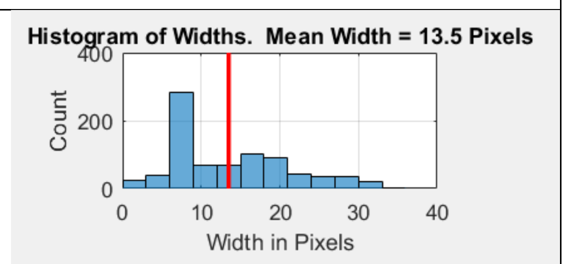
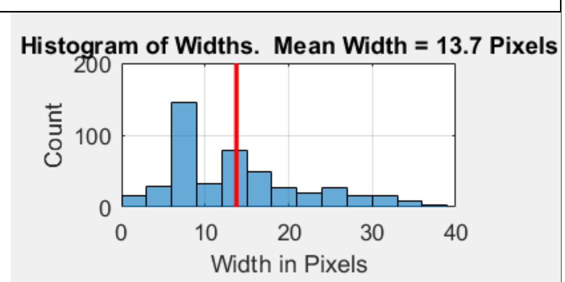
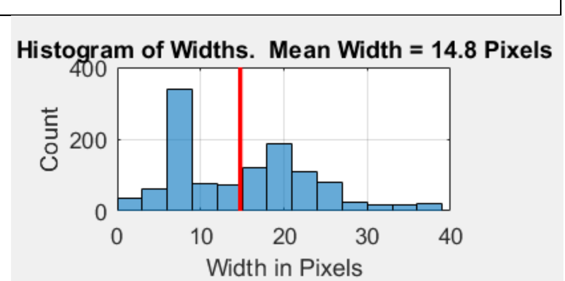
Image ID	Original Image	Ratio Value	Mean Value	Mean Histogram
CD-001		0.0240	13.7878	
CD-002		0.0350	7.8739	
CD-003		0.0486	14.5408	
CD-004		0.0495	18.3679	

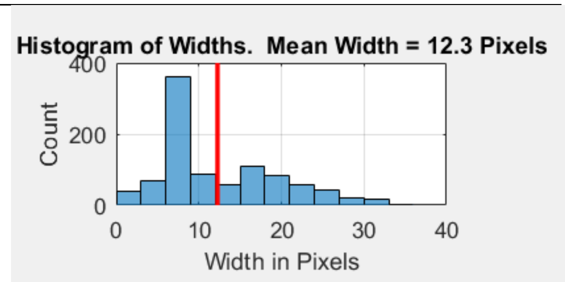
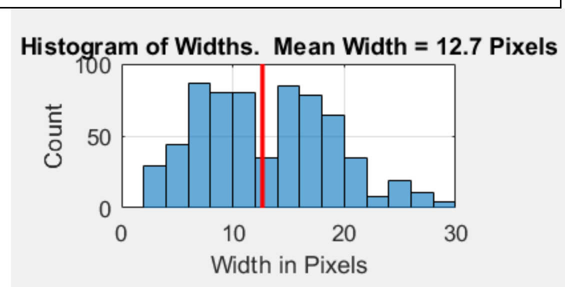
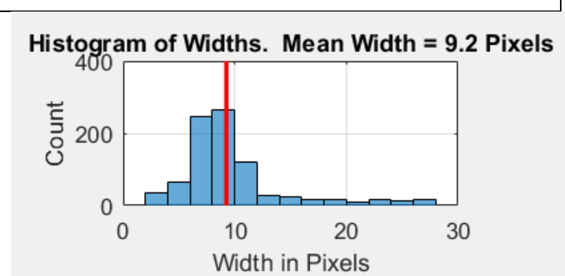
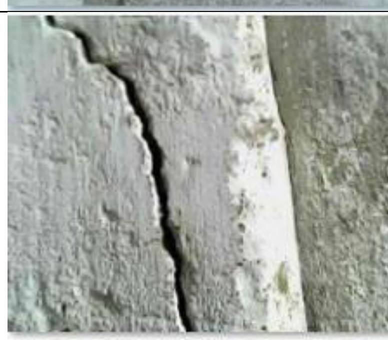
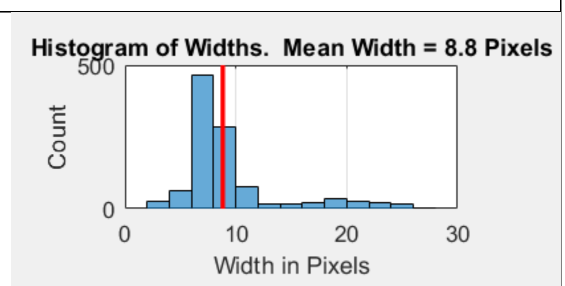
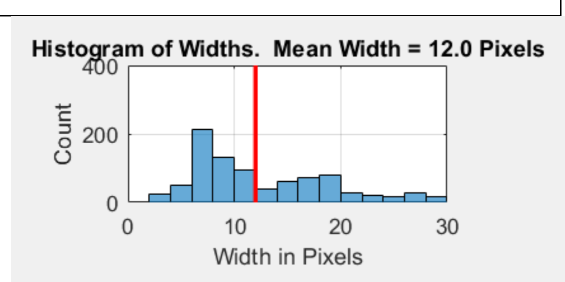
CD-005		0.0110	8.3583	
CD-006		0.0265	7.1759	
CD-007		0.0528	17.8112	
CD-008		0.1810	51.5778	
CD-009		0.0288	8.6519	

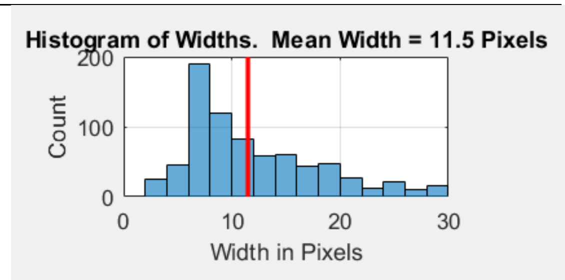
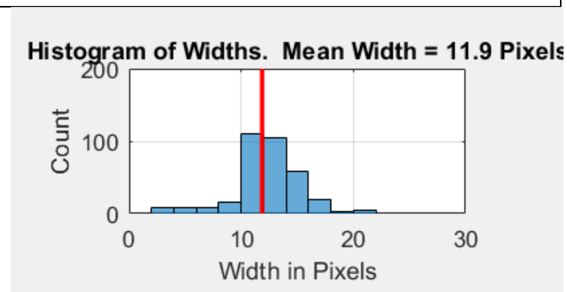
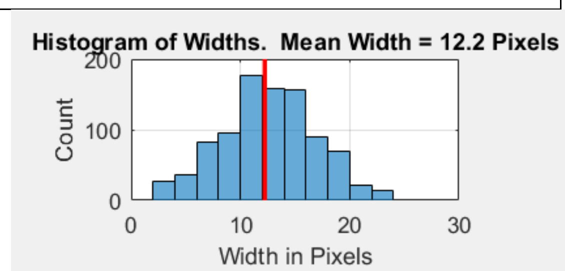
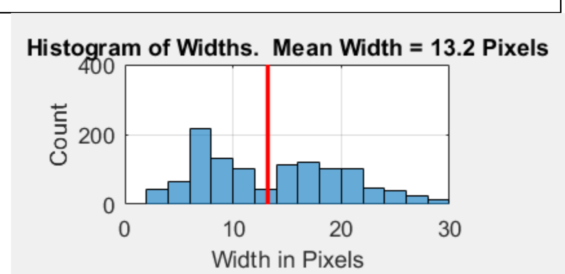
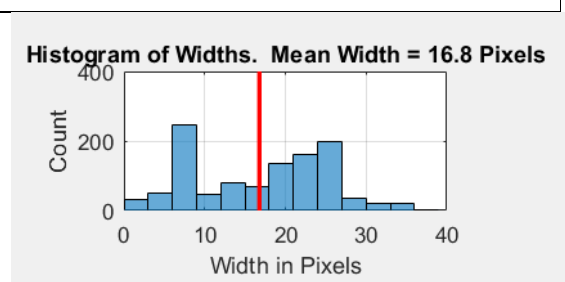
CD-010		0.0186	8.7177	
CD-011		0.0142	8.9782	
CD-012		0.0110	6.9933	
CD-013		0.0141	8.0681	
CD-014		0.0118	8.8673	

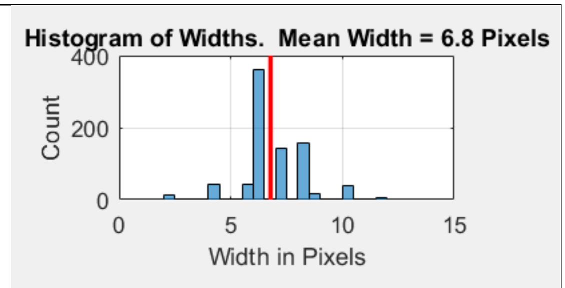
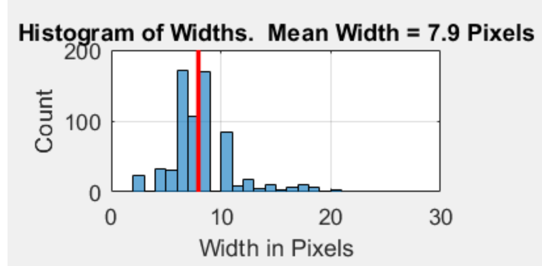
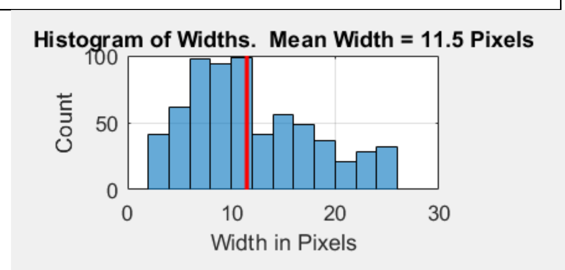
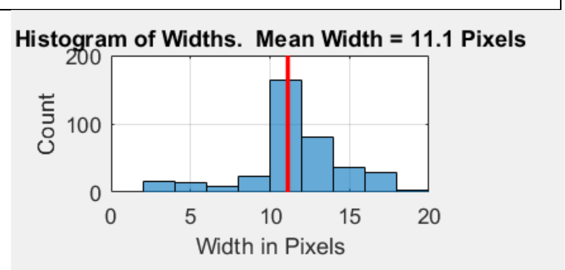
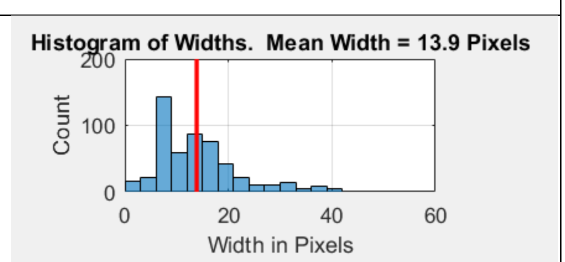
<table>
<tr><td>CD-015</td><td></td><td>0.0159</td><td>10.6901</td><td>
</td></tr>
<tr><td>CD-016</td><td></td><td>0.0485</td><td>13.2088</td><td>
</td></tr>
<tr><td>CD-017</td><td></td><td>0.0158</td><td>9.2027</td><td>
</td></tr>
<tr><td>CD-018</td><td></td><td>0.0284</td><td>11.4550</td><td>
</td></tr>
<tr><td>CD-019</td><td></td><td>0.0018</td><td>9.7386</td><td>
</td></tr>
</table>

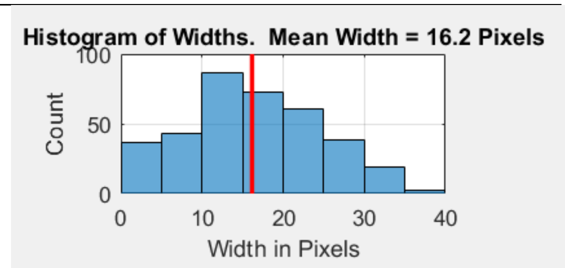
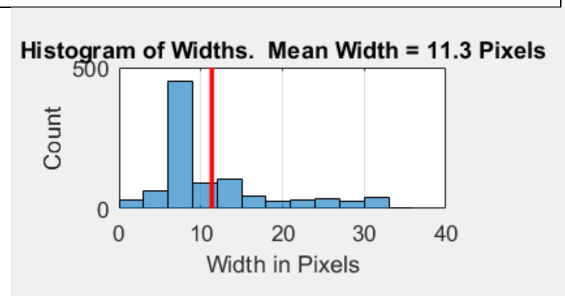
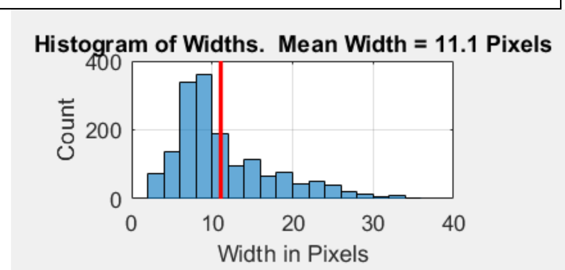
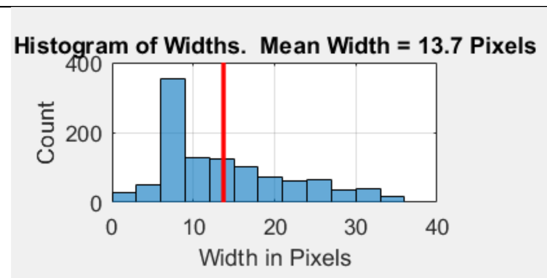
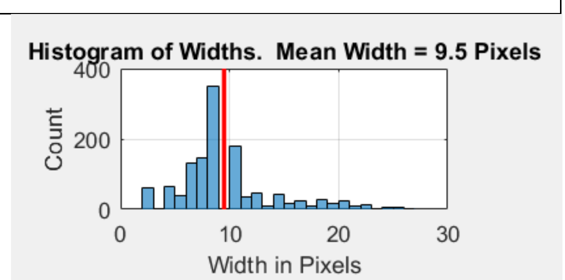
CD-020		0.0132	8.3717	
CD-021		0.0061	14.4544	
CD-022		0.0051	6.8958	
CD-023		0.0043	12.2258	
CD-024		0.0179	14.1322	

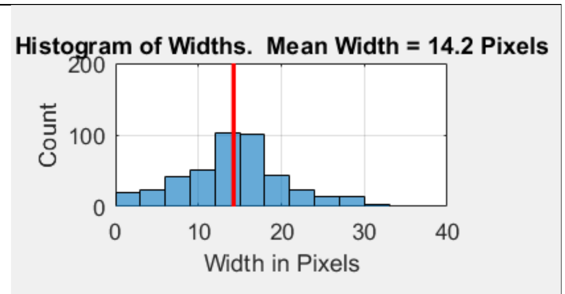
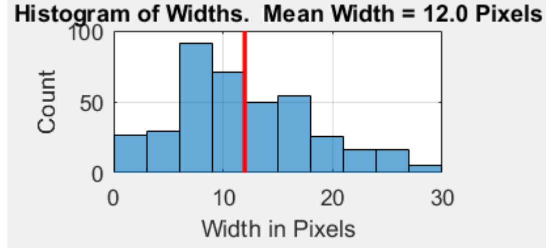
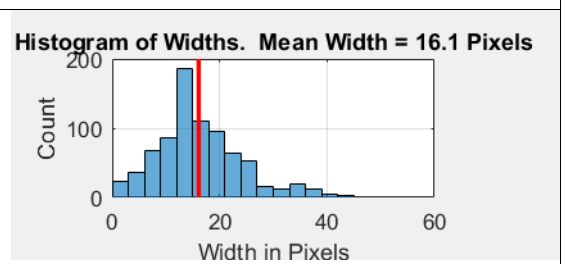
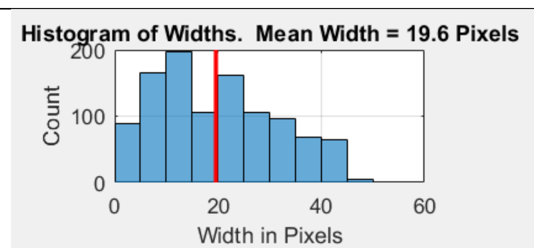
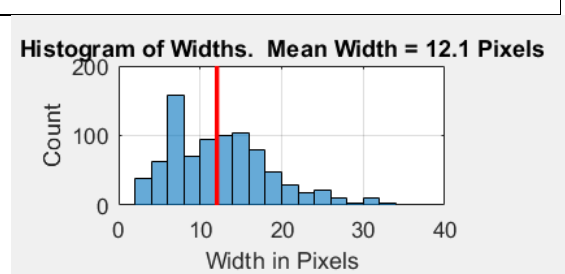
CD-025		0.0201	14.2081	
CD-026		0.0223	11.8742	
CD-027		0.0286	13.4959	
CD-028		0.0173	13.7462	
CD-029		0.0457	14.7628	

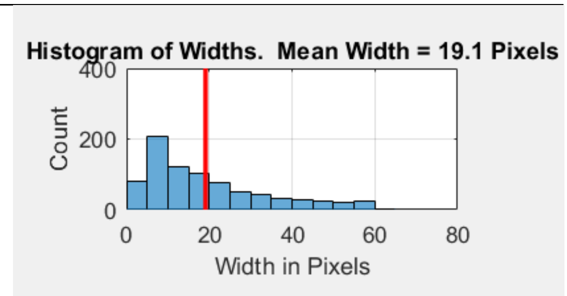
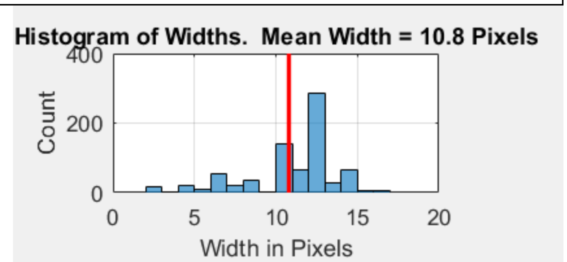
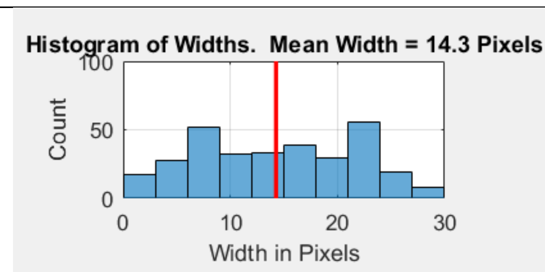
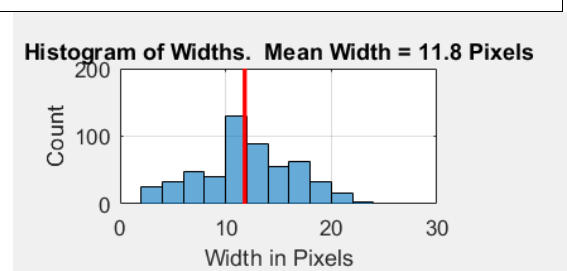
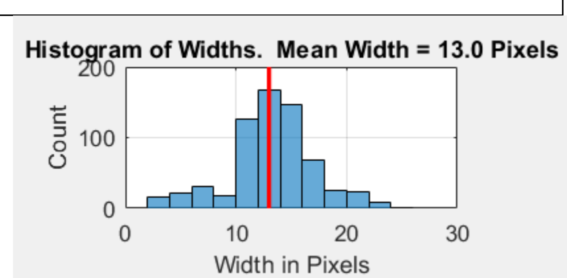
<table>
<tr><td>CD-030</td><td></td><td>0.0293</td><td>12.2582</td><td>
</td></tr>
<tr><td>CD-031</td><td></td><td>0.0228</td><td>12.6549</td><td>
</td></tr>
<tr><td>CD-032</td><td></td><td>0.0219</td><td>9.2480</td><td>
</td></tr>
<tr><td>CD-033</td><td></td><td>0.0264</td><td>8.8231</td><td>
</td></tr>
<tr><td>CD-034</td><td></td><td>0.0296</td><td>11.9908</td><td>
</td></tr>
</table>

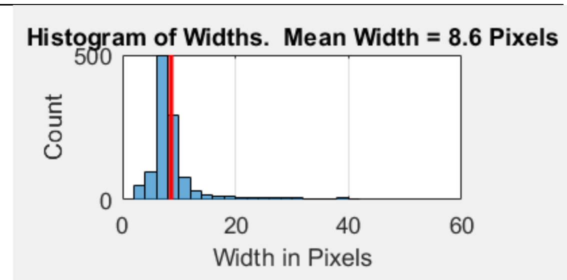
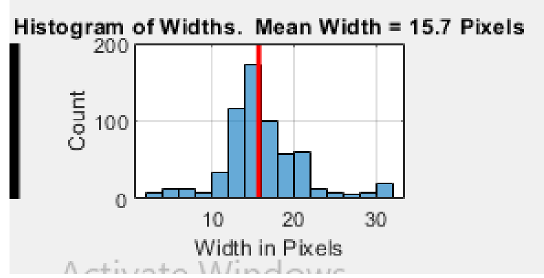
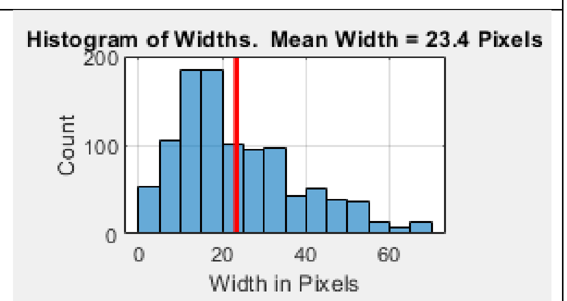
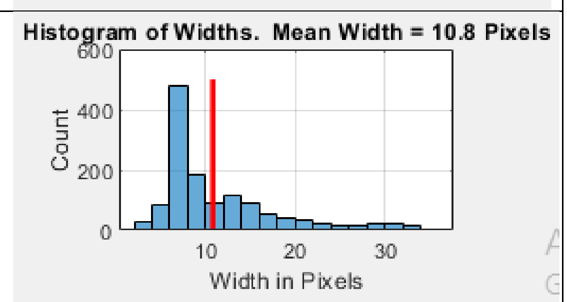
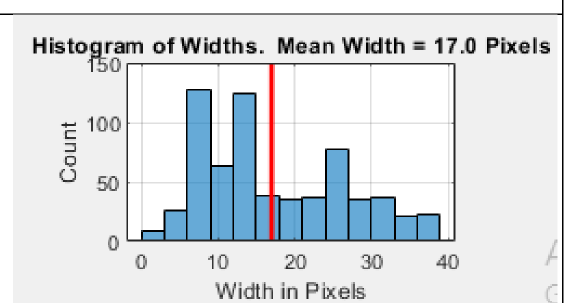
CD-035		0.0232	11.4841	
CD-036		0.0117	11.8610	
CD-037		0.0321	12.2331	
CD-038		0.0406	13.2078	
CD-039		0.0509	16.7830	

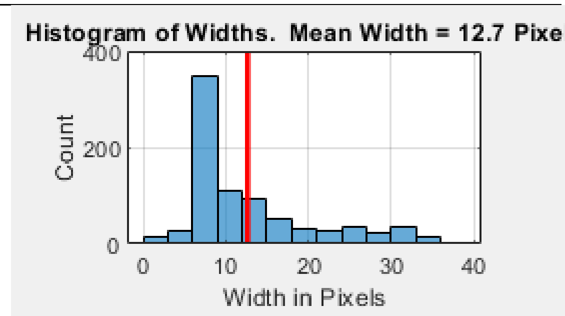
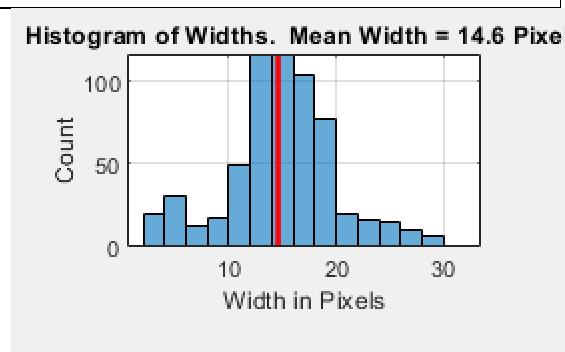
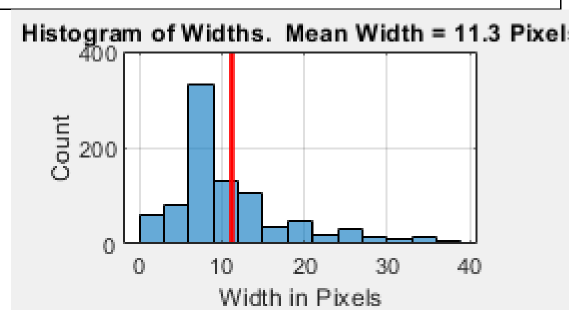
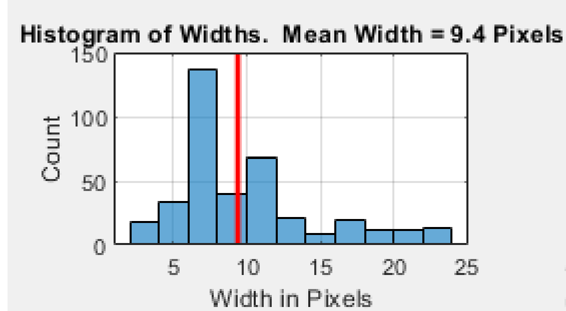
CD-040		0.0162	6.7770	
CD-041		0.0151	7.9412	
CD-042		0.0181	11.4875	
CD-043		0.0120	11.1192	
CD-044		0.0187	13.8794	

CD-045		0.0137	16.1601	
CD-046		0.0282	11.3482	
CD-047		0.0480	11.0789	
CD-048		0.0412	13.7153	
CD-049		0.0328	9.4987	

CD-050		0.0161	14.2411	
CD-051		0.0110	11.9653	
CD-052		0.0316	16.1184	
CD-053		0.0426	19.6020	
CD-054		0.0271	12.0670	

CD-055		0.0307	19.1134	
CD-056		0.0237	10.7975	
CD-057		0.0104	14.2657	
CD-058		0.0173	11.8039	
CD-059		0.0251	12.9850	

CD-060		0.0262	8.6052	
CD-061		0.0294	15.7036	
CD-062		0.0565	23.3583	
CD-063		0.0393	10.7959	
CD-064		0.0320	16.9966	

CD-065		0.0295	12.6585	
CD-066		0.0254	14.5957	
CD-067		0.0221	11.2539	
CD-068		0.0099	9.3867	

8.4.4 Detailed Table of Confusion matrix calculation

The table below shows the detailed process of calculations preceding confusion matrix generation. However, this dataset is limited to 45 images from the dataset.

	Image ID	Mean Value	Ratio Value	Predicted (Results)	Actual (Observed)	Cracked/ Non-Cracked	True/ False	Positive/ Negative	TP / FP / TN/ FN
1	CD-001	13.7878	0.0240171	Moderate	Moderate	Cracked	True	Positive	TP
2	CD-002	7.87387	0.349714	Moderate	Moderate	Cracked	True	Positive	TP
3	CD-003	14.5408	0.0486429	Moderate	Negligible	Cracked	False	Positive	FP
4	CD-004	18.3679	0.0495286	Negligible	Negligible	Cracked	True	Positive	TP
5	CD-005	8.35827	0.0109514	Negligible	Negligible	Cracked	True	Positive	FP
6	CD-007	17.8112	0.0527514	Severe	Moderate	Cracked	False	Positive	FP
7	CD-009	8.65193	0.0287543	Moderate	Moderate	Cracked	True	Positive	TP
8	CD-011	8.97821	0.01422	Moderate	Moderate	Cracked	True	Positive	TP
9	CD-012	6.99326	0.0110229	Negligible	Negligible	Cracked	True	Positive	TP
10	CD-013	8.06807	0.0141171	Moderate	Negligible	Cracked	False	Positive	FP
11	CD-015	10.6901	0.0158943	Moderate	Negligible	Cracked	False	Positive	FP
12	CD-017	17.0254	0.0349486	Negligible	Negligible	Cracked	True	Positive	TP
13	CD-018	12.2757	0.0128	Negligible	Negligible	Cracked	True	Positive	TP
14	CD-020	19.1616	0.0311371	Negligible	Negligible	Cracked	True	Positive	TP
15	CD-021	8.92539	0.0182657	Moderate	Moderate	Cracked	True	Positive	TP
16	CD-022	11.7016	0.0136429	Moderate	Negligible	Cracked	False	Positive	FP
17	CD-025	11.7961	0.0301343	Moderate	Moderate	Cracked	True	Positive	TP
18	CD-027	9.773865	0.00175143	Negligible	Negligible	Cracked	True	Positive	TP
19	CD-029	13.7509	0.0192171	Moderate	Negligible	Cracked	False	Positive	FP
20	CD-034	14.4544	0.00607143	Negligible	Negligible	Cracked	True	Positive	TP
21	CD-037	13.4959	0.02862	Moderate	Moderate	Cracked	True	Positive	TP
22	CD-040	14.3206	0.0549114	Negligible	Negligible	Cracked	True	Positive	TP
23	CD-042	11.6923	0.0260143	Moderate	Negligible	Cracked	False	Positive	FP
24	CD-045	10.7975	0.0237229	Moderate	Moderate	Cracked	True	Positive	TP
25	CD-046	20.314	0.0667343	Severe	Severe	Cracked	True	Positive	TP
26	CD-047	14.7075	0.03333	Moderate	Negligible	Cracked	False	Positive	FP
27	CD-048	8.60524	0.0261829	Moderate	Moderate	Cracked	True	Positive	TP
28	CD-050	12.985	0.0251371	Moderate	Negligible	Cracked	False	Positive	FP
29	CD-052	9.49873	0.0328057	Moderate	Moderate	Cracked	True	Positive	TP
30	CD-055	16.1184	0.0316343	Negligible	Negligible	Cracked	True	Positive	TP
31	CD-058	11.1192	0.0119857	Negligible	Negligible	Cracked	True	Positive	TP
32	CD-060	16.1601	0.0137486	Negligible	Negligible	Cracked	True	Positive	TP
33	CD-061	11.3482	0.0281514	Moderate	Moderate	Cracked	True	Positive	TP

34	CD-062	11.0789	0.0480029	Moderate	Moderate	Cracked	True	Positive	TP
35	CD-063	11.9908	0.0295686	Moderate	Moderate	Cracked	True	Positive	TP
36	CD-064	11.4841	0.0231514	Moderate	Moderate	Cracked	True	Positive	TP
37	CD-068	6.77701	0.0161857	Moderate	Moderate	Cracked	True	Positive	TP
38	CD-069	16.783	0.0508943	Severe	Severe	Cracked	True	Positive	TP
39	CD-071	14.7628	0.0456771	Moderate	Moderate	Cracked	True	Positive	TP
40	CD-073	8.8231	0.0264257	Moderate	Moderate	Cracked	True	Positive	TP
41	CD-074	12.6585	0.0295343	Moderate	Moderate	Cracked	True	Positive	TP
42	CD-075	10.8372	0.0245971	Moderate	Severe	Cracked	False	Positive	FP
43	CD-076	19.6437	0.0650029	Severe	Severe	Cracked	True	Positive	TP
44	CD-081	6.33477	0.00685714	Negligible	Slight	Non-Cracked	False	Negative	FN
45	CD-082	7.71412	0.00846857	Negligible	Slight	Non-Cracked	False	Negative	FN

8.4.5 Confusion Matrix Table and Performance Evaluator:

Number of Samples (n) = 250

TABLE 8.9: CONFUSION MATRIX TABLE

Predicted Class		**True Class**		
		Positive	**Negative**	**Total**
	Positive	183 (TP)	45 (FP)	228
	Negative	15 (FN)	7 (TN)	22
	Total	198	52	250

TABLE 8.10: PERFORMANCE EVALUATOR TABLE

Formula	Result (%)
Accuracy = $\frac{TP+TN}{TP+FP+FN+TN}$	76.0%
Precision = $\frac{TP}{TP+FP}$	80.0%
Sensitivity = $\frac{TP}{TP+FN}$	92.42%
Specificity = $\frac{TN}{FP+TN}$	13.46%
Positive predictive value = $\frac{TP}{TP+FP}$	80.0%
Negative predictive value = $\frac{TN}{FN+TN}$	31.81%

8.5 Bug List

Bug is an unexpected error of any software in this project. The bug list of the project is presented below which are likely to be encountered in the future but not necessarily. Bug bounty is a technique to remove bugs time to time

- Error appear after flight or images crashed, format bugs
- Database error, in case of maximum data sets (prediction), database bugs
- GUI scaling error, GUI bugs
- Program force shut down due to max burden, enforce bugs error
- User input error, input error bugs
- In case of auto optimizing of images and battery health face some loss and program close error, in handling bugs
- Close all pipes, security bugs error

Many bugs expected because every software have bugs, so the Bug bounty hunters find bugs and get rewards and enhance security system.

9. Conclusion

This project presents a crack detection technique for concrete structure based on the images of the concrete surface. This project will create safety and durability of the structure as cracks depict the need for repair priorities. The domain of this project is disaster management. Building crack detection is one of the time consuming as well as a hectic task for inspectors and manual process of crack detection for inspectors are also risky to reach high-rise buildings and detect building 'cracks. The main objective of the project was to classify the cracked surface and non-cracked surface. The drone camera will be programmed to obtain the required images/Video from a certain distance on its own. Furthermore, these images will be passed on to the Desktop Application and then find either there is cracked in the building or not, if cracks appear find how severe the crack was. Marked according to their severity.

Crack images from the drone, from a specified parameter, and image frames will be processed through an image processing desktop app using MATLAB programming. Images are analyzed by applying several image processing techniques and also applied different filters such as Canny, Sobel, and Prewitt. The canny filter gave the best result among these filters in our project. A good output result was being gained through adapting the strategy. The final output of the project is the same as what we aspect. However, despite the process of detection, the inspector has to visit locations for confirmation of the existence of those cracks and crack is detected.

10. Future Work

10.1 Introduction

The chapter discusses about the different approaches that follows the completion of this project. It explains the ways in which the image processing domain can be further explored and developed with the assistance of this project.

10.2 Future Work and Recommendations

Due to the time constraints, many other experiments and adaptations have been left for the future implementations. Firstly, future enhancements of this project can be to implement the 3D Model of the building after the successful detection of the cracks done through this project. It will further assist the inspectors in visualising the complete building in a 3D graphics which can easily identify all the cracks of the building at the same time and apply the process on the complete building instead of images. The scope of the project can also be extended such that coordinates of the buildings are pre-defined in the drone programming to automate the flying procedure of drone which captures images. Furthermore, the research conducted on the different research papers published by researchers to achieve the purpose of this research will also assist the researchers in the future. This project can also lead to a variety of different projects in the field of detection using the technique of image processing. Lastly, the admin panel for the head of the architect/inspector can also be added who can see the complete report of the building as well as add different users and provide them with privileges. Currently, the system is limited to the scope of a Desktop application; however, in future, the Mobile Application can also be developed to help the inspectors in an efficient way.

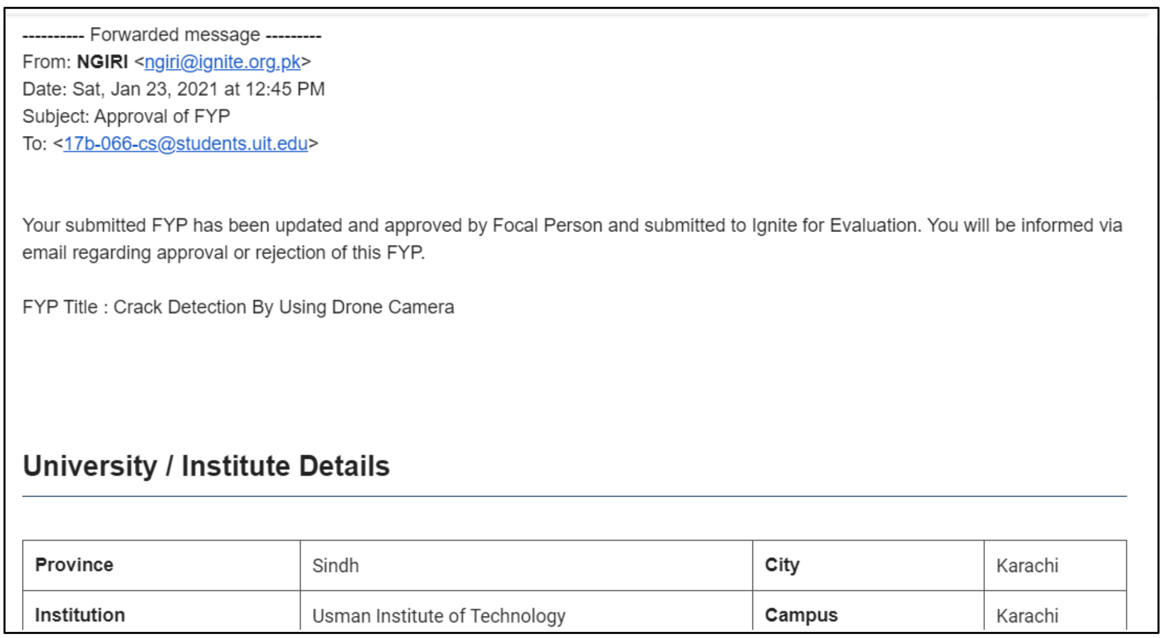
11. Achievements

11.1 Introduction

The chapter of Achievements pertains to describing the different participation of the proposed project in different exhibitions, incubation activities, and conferences. The participation of the current project in the funding process by ignite is described further in this chapter.

11.2 NGIRI by Ignite – National Technology Fund

The NGIRI program by Ignite was focused on providing financial assistance to the undergraduate students for the development of their Final Year Projects (FYP). It was aimed to ensure the development of skills and creativity among the individuals. This project was pitched and applied for funding in the NGIRI 2021. It was approved by the Focal Person and submitted for the evaluation by Ignite Foundation as shown in the Figure 11.1 below.



---------- Forwarded message ---------
From: **NGIRI** <ngiri@ignite.org.pk>
Date: Sat, Jan 23, 2021 at 12:45 PM
Subject: Approval of FYP
To: <17b-066-cs@students.uit.edu>

Your submitted FYP has been updated and approved by Focal Person and submitted to Ignite for Evaluation. You will be informed via email regarding approval or rejection of this FYP.

FYP Title : Crack Detection By Using Drone Camera

University / Institute Details

Province	Sindh	**City**	Karachi
Institution	Usman Institute of Technology	**Campus**	Karachi

FIGURE 11.1: APPLIED FOR NGIRI BY IGNITE

12. Appendices

12.1 Appendix A

TABLE 12.1: DETAILED SPECIFICATION OF THE DRONE CAMERA

S.no	Feature	Specification
1	Smartphone VR headset compatible for flexibility	Yes
2	Flight time	Above 9 minutes
3	Charging time	About 60 minutes
4	Weighs	0.08 kg
5	Frequency	2.4GHz
6	Drone Width	50 cm
7	Drone Height	19 cm
8	Drone Depth	50 cm
9	Battery Quantity Included (Drone)	1
10	Battery Cell Type (Drone)	7.4V 2000mAh Battery, Lithium Polymer

12.2 Appendix B

These are the out of scope objectives for the project

- This project will not work for real time covering building to detect cracks.
- This project will work around only some standards of cracks which is given above.

12.3 Appendix C

The deliverables of the project are described below:

- **Images/Video streaming**: Drone camera captures Images/Video streaming frames from certain distance.
- **Desktop Application**: Develop a desktop application providing the login/signup for the inspector, allowing him to detect cracks, view his profile and logout. The application will be showing the cracks marked according to their severity.

12.4 Appendix D

The estimated effort, cost and duration of the project is as follows:

Daily 7 to 8 hours and particular distribution of implementation testing and documentation, cost I think only depend on drone price only and duration is enough to given deadline to complete our project. The estimated effort hours are approximately 8 hours per day. On the other hand, the cost is associated with the system for running Desktop Application and purchasing of the drone camera only.

12.5 Appendix E

The estimated duration of the project is as follows:

Milestone	Date completed	Deliverable(s) completed
Project planning	11/18/20	• Crack Detection • Detect with drown
Milestone 1	12/27/20	• Algorithm • Crack Detection
Milestone 2	03/15/21	• Drone Programming
Milestone 3	06/1/21	• Testing
Milestone 4	06/11/21	• Final Product
Project conclusion	06/25/21	

12.6 Appendix F

There are no similar products available online, we found only similar research papers, and the URL list of some similar research papers is given below:

1. https://www.hindawi.com/journals/ace/2018/3924120/#literature-review
2. https://itc.scix.net/pdfs/w78-2014-paper-222.pdf
3. http://wcsit.org/media/pub/2018/vol.8.no.1/Crack%20Detection%20on%20Concrete%20Surfaces%20Using.pdf
4. https://www.sciencedirect.com/science/article/pii/S1110016817300236

13. References

[1] P. Dwivedi, "Detection of Surface Cracks in Concrete Structures using Deep Learning," *Medium*, 31-Dec-2019. [Online]. Available: https://towardsdatascience.com/detection-of-surface-cracks-in-concrete-structures-using-deep-learning-f8f85cd8ac8b. [Accessed: 10-Jan-2021].

[2] N.-D. Hoang, "Detection of Surface Crack in Building Structures Using Image Processing Technique with an Improved Otsu Method for Image Thresholding," *Advances in Civil Engineering*, vol. 2018, pp. 1–10, 2018.

[3] A. Mohan and S. Poobal, "Crack detection using image processing: A critical review and analysis," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 787–798, 2018.

[4] Y. Sato, Y. Bao, and Y. Koya, "Crack Detection on Concrete Surfaces Using V-shaped Features," *World of Computer Science and Information Technology Journal (WCSIT*, vol. 8, no. No 1, pp. 1–6, 2018.

[5] Dorafshan, Thomas, and Maguire, "Benchmarking Image Processing Algorithms for Unmanned Aerial System-Assisted Crack Detection in Concrete Structures," *Infrastructures*, vol. 4, no. 2, p. 19, 2019.

[6] J. Valença, I. Puente, E. Júlio, H. González-Jorge, and P. Arias-Sánchez, "Assessment of cracks on concrete bridges using image processing supported by laser scanning survey," Construction and Building Materials, vol. 146, pp. 668–678, 2017.

[7] X. Liu, Y. Ai, and S. Scherer, "Robust image-based crack detection in concrete structure using multi-scale enhancement and visual features," *2017 IEEE International Conference on Image Processing (ICIP)*, 2017.

[8] Y.-S. Yang, C.-L. Wu, T. T. Hsu, H.-C. Yang, H.-J. Lu, and C.-C. Chang, "Image analysis method for crack distribution and width estimation for reinforced concrete structures," *Automation in Construction*, vol. 91, pp. 120–132, 2018.

[9] Y. Wang, J. Y. Zhang, J. X. Liu, Y. Zhang, Z. P. Chen, C. G. Li, K. He, and R. B. Yan, "Research on Crack Detection Algorithm of the Concrete Bridge Based on Image Processing," *Procedia Computer Science*, vol. 154, pp. 610–616, 2019.

[10] D. Zhang, Q. Li, Y. Chen, M. Cao, L. He, and B. Zhang, "An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection," *Image and Vision Computing*, vol. 57, pp. 130–146, 2017.

[11] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," *2016 IEEE International Conference on Image Processing (ICIP)*, 2016.

[12] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, 2018.

[13] J. Zhu and J. Song, "An Intelligent Classification Model for Surface Defects on Cement Concrete Bridges," *Applied Sciences*, vol. 10, no. 3, p. 972, 2020.

[14] C. Su and W. Wang, "Concrete Cracks Detection Using Convolutional NeuralNetwork Based on Transfer Learning," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–10, 2020.

[15] Best Buy. DJI Ryze Tech Tello Quadcopter Drone with Camera white. Available Online on: https://www.bestbuy.ca/en-ca/product/dji-ryze-tech-tello-quadcopter-drone-with-camera-white/12644588

[16] DigiMax Pakistan. DJI Ryze Tech Tello Quadcopter. Available Online on: https://digimaxpakistan.com/product/dji-ryze-tech-tello-quadcopter/

[17] M. Larrea1, "Black-box testing technique for information visualization", Universidad Nacional del Sur, San Andres 800, Vol. 17 No. 1, April 2017.

[18] L. Pauly, H. Peel, S. Luo, D. Hogg, and R. Fuentes, "Deeper Networks for Pavement Crack Detection," *Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC)*, 2017.

[19] L. Some, "Automatic image-based road crack detection methods," *DIVA*, 01-Jul-2016. [Online]. Available: http://kth.diva-portal.org/smash/record.jsf?pid=diva2:945233&dswid=1383. [Accessed: 15-Sep-2021].

[20] S. Dorafshan, R. J. Thomas, and M. Maguire, "Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete," Construction and Building Materials, 11-Aug-2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950061818319500. [Accessed: 15-Sep-2021].

[21] C. V. Dung and L. D. Anh, "Autonomous concrete crack detection using deep fully convolutional neural network," *Automation in Construction*, 13-Dec-2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580518306745. [Accessed: 16-Sep-2021].

[22] M. M. M. Islam and J.-M. Kim, "Vision-Based Autonomous Crack Detection of Concrete Structures Using a Fully Convolutional Encoder–Decoder Network," *MDPI*, 30-Sep-2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/19/4251. [Accessed: 16-Sep-2021].

[23] A. N. Soni, “Crack Detection in buildings using convolutional neural ...” [Online]. Available: https://jidps.com/paper_issue/crack-detection-in-buildings-using-convolutional-neural-network/. [Accessed: 16-Sep-2021].

[24] E. Shahrokhinasab, N. Hosseinzadeh, A. Monirabbasi, and S. Torkaman, “Performance of Image-Based Crack Detection Systems in Concrete Structures,” *Journal of Soft Computing in Civil Engineering*, 01-Jan-2020. [Online]. Available: https://doi.org/10.22115/SCCE.2020.218984.1174. [Accessed: 16-Sep-2021].

[25] C. Zhang, X. Yu, L. Alexander, Y. Zhang, R. Rajamani, and N. Garg, “Piezoelectric active sensing system for crack detection in concrete structure,” *Journal of Civil Structural Health Monitoring*, 29-Feb-2016. [Online]. Available: https://link.springer.com/article/10.1007/s13349-015-0143-6. [Accessed: 16-Sep-2021].

[26] S. Cho, S. Park, G. Cha, and T. Oh, “Development of Image Processing for Crack Detection on Concrete Structures through Terrestrial Laser Scanning Associated with the Octree Structure,” *Applied Sciences*, vol. 8, no. 12, p. 2373, 2018.

[27] L. Wu, S. Mokhtari, A. Nazef, B. Nam, and H.-B. Yun, “Improvement of Crack-Detection Accuracy Using a Novel Crack Defragmentation Technique in Image-Based Road Assessment,” *Journal of Computing in Civil Engineering*, vol. 30, no. 1, p. 04014118, 2016.

[28] Ã. F. Ã–zgenel and A. G. Ã. Ã. SorguÃ§, “Performance Comparison of Pretrained Convolutional Neural Networks on Crack Detection in Buildings,” ISARC Proceedings, 22-Jul-2018. [Online]. Available:http://www.iaarc.org/publications/2018_proceedings_of_the_35th_isarc/performance_comparison_of_pretrained_convolutional_neural_networks_on_crack_detection_in_buildings.html. [Accessed: 16-Sep-2021].

[29] G.-L. Coca, Ș.-C. Romanescu, Ș.-M. Botez, and A. Iftene, “Crack detection system in

AWS Cloud using Convolutional neural networks," *Procedia Computer Science*, 02-Oct-2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920318652. [Accessed: 16-Sep-2021].

[30] P. Velumani, K. Mukilan, G. Varun, S. Divakar, R. M. Doss, and P. Ganeshkumar, "Analysis of cracks in structures and buildings," *Journal of Physics: Conference Series*, vol. 1706, p. 012116, 2020.

[31] Jonathan P. Rivera, Goran Josipovic, Emma Lejeune, Bismarck N. Luna, and Andrew S. Whittaker "Automated Detection and Measurement of Cracks in Reinforced Concrete Components" [Online] Available : http://sites.bu.edu/lejeunelab/files/2019/09/Automated_detection_and_measurement_of_cracks_in_reinforced_concrete_components.pdf [Accessed :16-sept-2021]

[32] S. Teng, Z. Liu, G. Chen, and L. Cheng, "Concrete Crack Detection Based on Well-Known Feature Extractor Model and the YOLO_v2 Network," *Applied Sciences*, vol. 11, no. 2, p. 813, 2021.

[33] Nawras Shatnawi Department of Surveying and Geomatics Engineering, Al-Balqa' Applied University, Al-Salt, Jorda [Online] Available : https://www.researchgate.net/publication/327983471_Automatic_Pavement_Cracks_Detection_using_Image_Processing_Techniques_and_Neural_Network/link/5bb22be7a6fdccd3cb811819/download [Accessed :16-sept-2021]

[34] Ravindra S. Hegadi " Image Processing: Research Opportunities and Challenges "[Online] Available : https://www.researchgate.net/publication/274247823_Image_Processing_Research_Opportunities_and_Challenges/link/551923f30cf273292e70a110/download

[35] Ajay Kumar Boyat and Brijendra Kumar Joshi " A REVIEW PAPER: NOISE

MODELS IN DIGITAL IMAGE PROCESSING”
[Online] Available: https://arxiv.org/ftp/arxiv/papers/1505/1505.03489.pdf

[36] Trần Duy Khánh “Image processing based building crack detection using Surface Crack Analysis program”[Online] Available: https://www.researchgate.net/publication/330079232_Image_processing_based_building_crack_detection_using_Surface_Crack_Analysis_program)

[37] Soojin Cho , Seunghee Park , Gichun Cha and Taekeun Oh “Development of Image Processing for Crack Detection on Concrete Structures through Terrestrial Laser Scanning Associated with the Octree Structure”
[Online] Available :https://www.researchgate.net/publication/329190245_Development_of_Image_Processing_for_Crack_Detection_on_Concrete_Structures_through_Terrestrial_Laser_Scanning_Associated_with_the_Octree_Structure

[38] Mrs.Preeti.Kale, Dr.S.T.Gandhe, Prof.Pravin.A.Dhulekar “ Enhancement of old images and documents by Digital Image Processing Techniques”
[Online] Available: https://www.researchgate.net/publication/281708917_Enhancement_of_old_images_and_documents_by_digital_image_processing_techniques/link/5f0c2852299bf188161987a6/download

[39] Ling Bai2 , Kuo Chen1 “Concrete surface crack detection with the improved pre-extraction and the second percolation processing methods” [Online] Available : https://storage.googleapis.com/plos-corpus-prod/10.1371/journal.pone.0201109/

[40] Xiaozhe Yuan “Study on Pavement Defect Detection Based on Image Processing Utilizing UAV” [Online] Available : https://www.researchgate.net/Comparison-on-the-Crack-Recognition-Effects-by-Different-Edge-Detection-Algorithms_tbl1_331679002

[41] K. Gopalakrishnan, “Deep Learning in Data-Driven Pavement Image Analysis and Automated Distress Detection: A Review,” *MDPI*, 24-Jul-2018. [Online]. Available: https://www.mdpi.com/2306-5729/3/3/28. [Accessed: 18-Sep-2021].

[42] L. Jiang, Y. Xie, and T. Ren, “A Deep Neural Networks Approach for Pixel-Level Runway Pavement Crack Segmentation Using Drone-Captured Images,” *NASA/ADS*. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2020arXiv200103257J/abstract. [Accessed: 18-Sep-2021].

[43] K. Chaiyasarn, “Crack Detection In Historical Structures Based On Convolutional Neural Network,” *International Journal of GEOMATE*, vol. 15, no. 51, 2018.

[44] A. Ragnoli, M. D. Blasiis, and A. D. Benedetto, “Pavement Distress Detection Methods: A Review,” *Infrastructures*, vol. 3, no. 4, p. 58, 2018.

[45] W. Cao, Q. Liu, and Z. He, “Review of Pavement Defect Detection Methods ,” *IEEE Access*, Jan. 2020. [Available Online] at: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8960347https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8960347

[46] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou, and X. Shen, “Image-based concrete crack detection in tunnels using deep fully convolutional networks,” *Construction and Building Materials*, vol. 234, p. 117367, 2020.

[47] H. K. Jung and G. Park, “Rapid and non-invasive surface crack detection for pressed-panel products based on online image processing,” *Structural Health Monitoring*, vol. 18, no. 5-6, pp. 1928–1942, 2019.

[48] J. Akbari, M. Ahmadifarid, and A. K. Amiri, “Multiple Crack Detection using Wavelet Transforms and Energy Signal Techniques,” *Frattura ed Integrità Strutturale*, vol. 14,

no. 52, pp. 269–280, 2020.

[49] B. Romero-Tarazona, C. Rodriguez-Sandoval, J. Villabonai-Ascanio, and A. D. Rincón-Quintero, “Development of an artificial vision system that allows non-destructive testing on flat concrete slabs for surface crack detection by processing of digital images in MATLAB,” *IOP Conference Series: Materials Science and Engineering*, vol. 844, p. 012058, 2020.

[50] Q. Gong, L. Zhu, Y. Wang, and Z. Yu, “Automatic subway tunnel crack detection system based on line scan camera,” *Structural Control and Health Monitoring*, 2021.

14. Plagiarism Report

CS 16

ORIGINALITY REPORT

8%	6%	4%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	en.wikipedia.org Internet Source	2%
2	www.mdpi.com Internet Source	<1%
3	www.coursehero.com Internet Source	<1%
4	Submitted to University of Greenwich Student Paper	<1%
5	towardsdatascience.com Internet Source	<1%
6	thunder.state.sd.us Internet Source	<1%
7	repositorio.ufpe.br Internet Source	<1%
8	Submitted to Liverpool John Moores University Student Paper	<1%
9	www.hindawi.com Internet Source	<1%

10	www.semanticscholar.org Internet Source	<1 %
11	Submitted to Technische Universiteit Delft Student Paper	<1 %
12	Yun Wang, Ju Yong Zhang, Jing Xin Liu, Yin Zhang, Zhi Ping Chen, Chun Guang Li, Kai He, Rui Bin Yan. "Research on Crack Detection Algorithm of the Concrete Bridge Based on Image Processing", Procedia Computer Science, 2019 Publication	<1 %
13	Submitted to St. Peters Lutheran College Student Paper	<1 %
14	Submitted to University of Western Sydney Student Paper	<1 %
15	doaj.org Internet Source	<1 %
16	downloads.hindawi.com Internet Source	<1 %
17	graph.architexturez.net Internet Source	<1 %
18	Submitted to The University of Manchester Student Paper	<1 %
19	pt.scribd.com Internet Source	<1 %

20	repository.essex.ac.uk Internet Source	<1 %
21	download.wcsit.org Internet Source	<1 %
22	eprints.usq.edu.au Internet Source	<1 %
23	www.bestbuy.ca Internet Source	<1 %
24	www.pbtech.co.nz Internet Source	<1 %
25	P Velumani, K Mukilan, G Varun, S Divakar, R Muhil Doss, P Ganeshkumar. "Analysis of cracks in structures and buildings", Journal of Physics: Conference Series, 2020 Publication	<1 %
26	Arun Mohan, Sumathi Poobal. "Crack detection using image processing: A critical review and analysis", Alexandria Engineering Journal, 2017 Publication	<1 %
27	researchr.org Internet Source	<1 %
28	Jonathan P. Rivera, Goran Josipovic, Emma Lejeune, Bismarck N. Luna, Andrew S. Whittaker. "Automated Detection and Measurement of Cracks in Reinforced	<1 %

Concrete Components", ACI Structural Journal, 2015
Publication

29	Submitted to University of Westminster Student Paper	<1 %
30	Zhang, Dejin, Qingquan Li, Ying Chen, Min Cao, Li He, and Bailing Zhang. "An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection", Image and Vision Computing, 2017. Publication	<1 %
31	Submitted to University of Leicester Student Paper	<1 %
32	Submitted to University of Southampton Student Paper	<1 %
33	Nhat-Duc Hoang. "Detection of Surface Crack in Building Structures Using Image Processing Technique with an Improved Otsu Method for Image Thresholding", Advances in Civil Engineering, 2018 Publication	<1 %
34	Tarun Jain, Vivek Kumar Verma, Payal Garg, Mahesh Jangid. "Chapter 11 An Improvised Model for High-Security License Plate Detection and Recognition for Indian Vehicle to Enhance Detection Accuracy", Springer Science and Business Media LLC, 2020 Publication	<1 %

35	ijarcet.org Internet Source	<1%
36	www.ijritcc.org Internet Source	<1%
37	Submitted to Birkbeck College Student Paper	<1%
38	A. Diana Andrushia, N. Anand, G. Prince Arulraj. "Evaluation of thermal cracks on fire exposed concrete structures using ripplet transform", Mathematics and Computers in Simulation, 2020 Publication	<1%
39	Aishwarya Aishwarya, Parth Wadhwa, Owais Owais, Vasudha Vashisht. "A Waste Management Technique to detect and separate Non-Biodegradable Waste using Machine Learning and YOLO algorithm", 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021 Publication	<1%
40	Arathi Reghukumar, L. Jani Anbarasi. "Chapter 19 Crack Detection in Concrete Structures Using Image Processing and Deep Learning", Springer Science and Business Media LLC, 2021 Publication	<1%

41	Fen Fang, Liyuan Li, Ying Gu, Hongyuan Zhu, Joo-Hwee Lim. "A novel hybrid approach for crack detection", Pattern Recognition, 2020 Publication	<1 %
42	Submitted to Higher Education Commission Pakistan Student Paper	<1 %
43	Wenting Qiao, Hongwei Zhang, Fei Zhu, Qiande Wu. "A Crack Identification Method for Concrete Structures Using Improved U-Net Convolutional Neural Networks", Mathematical Problems in Engineering, 2021 Publication	<1 %
44	coek.info Internet Source	<1 %
45	espace.library.uq.edu.au Internet Source	<1 %
46	pemrogramanmatlab.com Internet Source	<1 %
47	Avik Kumar Das, Christopher.K.Y. Leung, Kai Tai Wan. "Application of deep convolutional neural networks for automated and rapid identification and computation of crack statistics of thin cracks in strain hardening cementitious composites (SHCCs)", Cement and Concrete Composites, 2021 Publication	<1 %

48	Esraa Elhariri, Nashwa El-Bendary, Shereen A. Taie. "Using Hybrid Filter-Wrapper Feature Selection with Multi-Objective Improved-Salp Optimization for Crack Severity Recognition", IEEE Access, 2020 Publication	<1 %
49	Hyunjin Bae, Keunyoung Jang, Yun-Kyu An. "Deep super resolution crack network (SrcNet) for improving computer vision–based automated crack detectability in in situ bridges", Structural Health Monitoring, 2020 Publication	<1 %
50	J. Valença, I. Puente, E. Júlio, H. González-Jorge, P. Arias-Sánchez. "Assessment of cracks on concrete bridges using image processing supported by laser scanning survey", Construction and Building Materials, 2017 Publication	<1 %
51	N Aravind, S Nagajothi, S Elavenil. "Machine learning model for predicting the crack detection and pattern recognition of geopolymer concrete beams", Construction and Building Materials, 2021 Publication	<1 %
52	Nhat-Duc Hoang, Quoc-Lam Nguyen. "A novel method for asphalt pavement crack classification based on image processing and	<1 %

machine learning", Engineering with Computers, 2018
Publication

53	Norman F. Schneidewind. "Integrating Testing with Reliability", Wiley, 2011 Publication	<1 %
54	Richard Amankwah, Jinfu Chen, Patrick Kwaku Kudjo, Beatrice Korkor Agyemang, Alfred Adutwum Amponsah. "An automated framework for evaluating open-source web scanner vulnerability severity", Service Oriented Computing and Applications, 2020 Publication	<1 %
55	Wenming Cao, Qifan Liu, Zhiquan He. "Review of Pavement Defect Detection Methods", IEEE Access, 2020 Publication	<1 %
56	Yanli Wang, Yuchun Huang, Weihong Huang. "Crack Junction Detection in Pavement Image Using Correlation Structure Analysis and Iterative Tensor Voting", IEEE Access, 2019 Publication	<1 %
57	Yue Pan, Gaowei Zhang, Limao Zhang. "A spatial-channel hierarchical deep learning network for pixel-level automated crack detection", Automation in Construction, 2020 Publication	<1 %

58	Ziping Gao, Bo Peng, Tianrui Li, Cong Gou. "Generative Adversarial Networks for Road Crack Image Segmentation", 2019 International Joint Conference on Neural Networks (IJCNN), 2019 Publication	<1 %
59	docplayer.net Internet Source	<1 %
60	ir.lib.uwo.ca Internet Source	<1 %
61	strathprints.strath.ac.uk Internet Source	<1 %
62	www.ijrte.org Internet Source	<1 %
63	"Advances in Electrical and Computer Technologies", Springer Science and Business Media LLC, 2021 Publication	<1 %
64	Eslam Mohammed Abdelkader, Osama Moselhi, Mohamed Marzouk, Tarek Zayed. "A Grey Wolf Optimization-Based Method for Segmentation and Evaluation of Scaling in Reinforced Concrete Bridges", International Journal of Information Technology & Decision Making, 2021 Publication	<1 %

65	Jinsong Zhu, Chi Zhang, Haidong Qi, Ziyue Lu. "Vision-based defects detection for bridges using transfer learning and convolutional neural networks", Structure and Infrastructure Engineering, 2019 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		