[ANALYSIS ON DIGITAL IMAGE CORRELATION USING IMAGE PROCESSING]

[ANISAH BINTI KAMSIN]

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS LAPORAN

JUDUL: [ANALYSIS ON DIGITAL IMAGE CORRELATION USING IMAGE PROCESSING]

SESI PENGAJIAN: [2021 / 2022]

Saya: [ANISAH BINTI KAMSIN]

mengaku membenarkan tesis Projek Sarjana Muda ini disimpan di Perpustakaan Universiti Teknikal Malaysia Melaka dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan unituk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. * Sila tandakan (✓)

______	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
______	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)
___✓___	TIDAK TERHAD	

Anisah Kamsin

(TANDATANGAN PELAJAR)

Alamat tetap: JA 9108, Jalan Debunga 6, Taman Debunga, 77300, Merlimau, Melaka

Tarikh: 10 September 2022

(TANDATANGAN PENYELIA)

DR. ZURAINI BINTI OTHMAN

Tarikh: 10 September 2022

[ANALYSIS ON DIGITAL IMAGE CORRELATION USING IMAGE PROCESSING]

[ANISAH BINTI KAMSIN]

This report is submitted in partial fulfillment of the requirements for the Bachelor of [Computer Science (Artificial Intelligence)] with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2022

DECLARATION

I hereby declare that this project report entitled

[ANALYSIS ON DIGITAL IMAGE CORRELATION USING IMAGE PROCESSING]

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : ANISAH BINTI KAMSIN Date : 10/9/2022

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of [Computer Science (Artificial Intelligence)] with Honours.

SUPERVISOR : DR. ZURAINI BINTI OTHMAN Date: 10/9/2022

DEDICATION

I humbly dedicate this Final Year Project (FYP) to my beloved parents who were constantly there for me throughout the ups and downs of life and for showering me with endless love and affection. Not to forget my supervisor, Dr Zuraini Binti Othman, for continuously guiding me and bringing out the best in me. Besides, I dedicate this project to my friend, Fairuz Diana Binti Hamzah who has always given me encouragement when I felt hopeless.

ACKNOWLEDGEMENTS

First and foremost, all praises to Allah S.W.T the Almighty, for His showers of blessing, strength, and light of guidance throughout completing this project successfully.

Next, I would like to express my deepest gratitude and appreciation to my supervisor, Dr. Zuraini Binti Othman, for giving me this invaluable opportunity to conduct this project, and providing motivations and constructive criticisms for me to strive better in this project.

Last but not least, this research would not have been possible without the endless supports, prayers, and assurance from my family members, and my friends.

ABSTRACT

The recently created Digital Image Correlation (DIC) technique is an image identification method that can be used to measure object deformation. The deformation and strain field of an object can be calculated using the point on the image using this technique, which also has the ability to correlate the digital images of an object both before and after deformation. Usually this method is used in engineering using Ncorr (MATLAB) only. In this paper, the discovery of strain is not what we want to find, but analysis of the DIC technique using image processing because there is not much research about image processing being applied to the DIC technique and it is hard for other researchers to do analysis on it. So, the significance of this project is by using the proposed algorithm, developer can implement this algorithm in their application which greatly benefit the community in providing information about the DIC. This project is also carried out through MATLAB software. The image dataset used here is a transformation from video to still images in sequence where a high-speed digital camera used must be stationary and stable in order to record or capture multiple images of material/specimen deformation. This will lead to the discovery of a more precise value. The image data set is then undergoing image processing to obtain a more meaningful picture quality. The result of the correlation values can be defined from the post-process image and after the values are plotted, there are an increase and decrease until, on the 7th image, the correlation value becomes the highest value. It shows that there is an expansion of the molecules where it causes the specimen to break.

ABSTRAK

Teknik Korelasi Imej Digital (DIC) yang dibuat baru-baru ini ialah kaedah pengenalan imej yang boleh digunakan untuk mengukur ubah bentuk objek. Medan ubah bentuk dan terikan sesuatu objek boleh dikira menggunakan titik pada imej menggunakan teknik ini, yang juga mempunyai keupayaan untuk mengaitkan imej digital sesuatu objek sebelum dan selepas ubah bentuk. Biasanya kaedah ini digunakan dalam bidang kejuruteraan menggunakan Ncorr (MATLAB) sahaja. Dalam kertas kerja ini, penemuan ketegangan bukanlah apa yang ingin dicari, tetapi analisis teknik DIC menggunakan pemprosesan imej kerana tidak banyak kajian tentang pemprosesan imej yang digunakan untuk teknik DIC dan sukar untuk penyelidik lain membuat analisis keatasnya atasnya. Jadi, kepentingan projek ini adalah dengan menggunakan algoritma yang dicadangkan, pembangun boleh melaksanakan algoritma ini dalam aplikasi mereka yang memberi manfaat yang ketara kepada komuniti dalam memberikan maklumat tentang DIC. Projek ini juga dijalankan melalui perisian MATLAB. Set data imej yang digunakan di sini ialah transformasi daripada video kepada imej pegun dalam urutan di mana kamera digital berkelajuan tinggi yang digunakan mestilah pegun dan stabil untuk merakam atau menangkap berbilang imej ubah bentuk bahan/spesimen. Ini akan membawa kepada penemuan nilai yang lebih tepat. Set data imej kemudiannya menjalani pemprosesan imej untuk mendapatkan kualiti gambar yang lebih bermakna. Hasil daripada nilai korelasi boleh ditakrifkan daripada imej pasca proses dan selepas nilai diplot, terdapat peningkatan dan penurunan sehingga, pada imej ke-7, nilai korelasi menjadi nilai tertinggi. Ia menunjukkan bahawa terdapat pengembangan molekul di mana ia menyebabkan spesimen pecah.

TABLE OF CONTENTS

PAGE

DECLARATION II

DEDICATION III

ACKNOWLEDGEMENTS IV

ABSTRACT V

ABSTRAK VI

TABLE OF CONTENTS VII

LIST OF TABLES X

LIST OF FIGURES XI

LIST OF ABBREVIATIONS XIII

LIST OF ATTACHMENTS XIV

CHAPTER 1: INTRODUCTION 1

1.1 Introduction 1

1.2 Problem statement(s) 2

1.3 Objective 3

1.4 Scope 4

1.5 Expected output 4

1.6 Conclusion 4

CHAPTER 2: LITERATURE REVIEW 5

2.1 Introduction 5

2.1.1 Different pattern used as reference entities of measurement 5

2.2 Related Work/Previous Work 6

2.3 Critical review of current problem and justification 11

2.4 Proposed Solution/further project 12

2.5 Conclusion 12

CHAPTER 3: PROJECT METHODOLOGY 13

3.1 Introduction 13

3.2 Methodology 13

3.2.1 Convert image to gray image 14

3.2.2 Noise removal 14

3.2.3 Contrast enhancement 15

3.2.4 Calculate Mean Grayscale 16

3.2.5 Correlation Values 18

3.2.6 Area of Mean Grayscale 19

3.2.7 Correlation Graph 20

3.3 Project Milestones 20

3.4 Conclusion 21

CHAPTER 4: DESIGN 22

4.1 Introduction 22

4.2 Network System Architecture 22

4.3 Possible Scenarios 23

4.4 Metric Measurement 26

4.5 Conclusion 29

CHAPTER 5: IMPLEMENTATION 30

5.1 Introduction 30

5.2 Environment Setup 30

5.3 Conclusion 32

CHAPTER 6: TESTING AND ANALYSIS 33

6.1 Introduction 33

6.2 Results and Analysis 33

6.3 Conclusion 35

CHAPTER 7: PROJECT CONCLUSION 36

7.1 Introduction 36

7.2 Project Summarization 36

7.3 Project Contribution 36

7.4 Project Limitation 36

7.5 Future Works 37

7.6 Conclusion 37

CHAPTER 8: REFERENCES 38

APPENDIX 40

LIST OF TABLES

PAGE

Table 1: Review existing research paper 8
Table 2: PSM 1 21
Table 3: PSM 2 21

LIST OF FIGURES

PAGE

Figure 1: UTM machine 2
Figure 2: Conventional Strain Gauges 3
Figure 3: An example of a random speckle pattern. 6
Figure 4: An example of an ROI and a grid of markers. 6
Figure 5: Flowchart of analysis DIC 13
Figure 6: Example of Noise Removal 15
Figure 7: Example of Contrast Enhancement 15
Figure 8: Cropped Image 16
Figure 9: Coefficients 17
Figure 10: Background Corrected Image 17
Figure 11: Graph of Width 18
Figure 12: How to find correlation values 18
Figure 13: Graph of mean grayscale of the background corrected images 19
Figure 14: Area of Mean Grayscale Graph 19
Figure 15: Correlation Graph 20
Figure 16: Data Flow Diagram of the system 22
Figure 17: Read and convert to Gray Image 23
Figure 18: Noise Removal 23
Figure 19: Contrast Enhancement 23
Figure 20: Preprocessing Image 24
Figure 21: How to crop the desired segment 24
Figure 22: Display cropped image 25
Figure 23: Result of displaying cropped image 25
Figure 24: Calculate the coefficients of the image 26

Figure 25: Plot the value to the graph 26
Figure 26: Graph of coefficients of the image 27
Figure 27: Create a background image 27
Figure 28: Display the background corrected image 28
Figure 29: Results of background corrected image 28
Figure 30: The Mean Grayscale codes 28
Figure 31: Graph of width 29
Figure 32: App Designer 30
Figure 33: Code of Browse Button pushed function 31
Figure 34: Code of Processing Button pushed function 31
Figure 35: Code of ROI Button pushed function 32
Figure 36: Code of BCI Button pushed function 32
Figure 37: GUI of DIC technique using Image Preprocessing 33
Figure 38: Correlation Graph 34
Figure 39: Images Vs Correlations 34
Figure 40: Zoom in to Image 7 where it starts to break 35

LIST OF ABBREVIATIONS

GUI	**-**	**Graphical User Interface**
DIC	**-**	**Digital Image Correlation**
PIV	**-**	**Particle Image Velocimetry**
DAQ	**-**	**Data Acquisition System**
ROI	**-**	**Region of Interest**
BCI	**-**	**Background Corrected Image**
UTM	**-**	**Universal Tensile Machine**

LIST OF ATTACHMENTS

		PAGE
Appendix A	**Source Codes**	**36**
Appendix B	**Analysis of Data**	**57**
Appendix C	**Image Dataset Sample 2**	**64**
Appendix D	**Output for Image Dataset Sample 2**	**64**
Appendix E	**Plagiarism**	**65**

CHAPTER 1: INTRODUCTION

1.1 Introduction

Digital Image Correlation (DIC) consists in registering two (or more) images of the same scene and extracting the displacement fields that enable obtaining the best match. DIC, which is used in solid mechanics, is the counterpart of particle image velocimetry (PIV) used in fluid mechanics. DIC was initiated a few years later than PIV. However, because the sought displacement and strain resolutions were smaller than the velocity resolutions, its development was slower because the algorithmic challenges were more difficult to address. Today, it can be stated that the performance of DIC techniques allows the experimentalist to utilize them in most practical cases.

In many engineering applications, the measurement of displacement or deformation at any point in a structural element or material subject to external loads is important. The strength issues in a structural member are visualised using the obtained strain values. Strain gauges and extensometers, which are common strain measurement devices, are unable to produce strain maps. The Digital Image Correlation (DIC) method enables a full-field strain measurement. This technique is a member of the class of optical, non-interferometric methods that assess deformation by contrasting variations in the surface image of a test object before and after deformation. The DIC method enables novel and intricate investigations. (Jarosław Górszczyk,[*] Konrad Malicki, and Teresa Zych, 2019)

The function of image processing can be categorized into the following four categories, preprocessing, image enhancement, image transformation, and image classification. After one or all process has been done on the image, it will make the image quality better than the raw/original image where it can be used for the purpose of pixel image analysis and so on. DIC is an optical technique that uses tracking and image registration methods to measure changes in images accurately in 2D and 3D. To evaluate full-field displacement and strains, this technique is frequently used. So, with the help of image processing, it will be able to provide accurate measurements in a study.

The DIC technique enables a full-field strain measurement. This technique is one of a group of optical, non-interferometric methods that assesses deformation by contrasting variations in the surface image of an object under test before and after deformation. In general, DIC tracks the movement of pixels in the region of interest (ROI), compares a sequence of grey-scale images of a specimen at various stages of deformation, and uses a correlation algorithm to calculate displacement and strain. (SenLin, 2015)

1.2 Problem statement(s)

Digital image correlation method is being widely used for displacement measurement in various areas of science and engineering. Figure 1 below shows a high speed digital camera where it will be used to record or capture multiple images of the material/specimen deformation. However, it can be less accurate than traditional testing, especially if the camera is hit or moved during testing (Jake Daniels and Travis Manning).

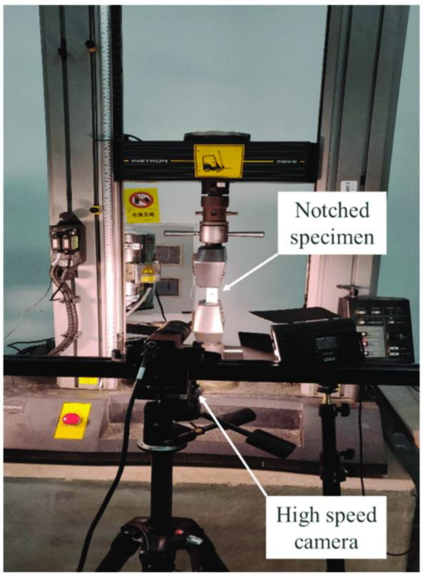


Figure 1: UTM machine

They have also said the traditional testing is not recommended because many additional procedures are required when utilizing a physical strain gauge. This testing is called conventional strain gauges (see Figure 2).

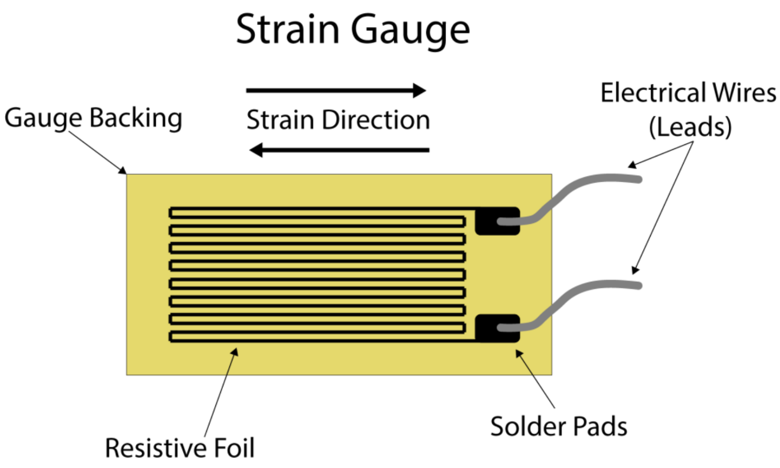


Figure 2: Conventional Strain Gauges

First and foremost, these gauges necessitate the removal of the bridge's protective coating, which might irreparably harm the bridge. Furthermore, the foil gauges require a significant amount of wiring that must be joined to a Data Acquisition system (DAQ). This DAQ must also be linked to a computer, resulting in a large amount of equipment on site.

1.3 Objective

Aim of this project is:

1. To explore and compare the recent research.
2. To propose a Digital Image Correlation framework.
3. To obtain the accurate analysis of the material being stressed under load.

1.4 Scope

Scope of this project is:

1. Captured images.
2. Compare the proposed technique with the recent projects.

1.5 Expected output

Expected output of this project is able to investigate and evaluate the most recent research, successfully suggest an image processing-based framework for digital image correlation, and get to assess the accuracy of the analysis on the material that is under stress.

1.6 Conclusion

As a summary, this chapter consists of basic understandings about the DIC technique. It also contains the project background, problem statements, objectives, and scope of the project. In the next chapter, research and projects related to this project that has been done by other researchers will be discussed in detail.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

The literature review of this project focuses more on results and discussions of several research papers and journals. The fundamental reason for this literature review and investigation is to direct the project with reference to substantial and dependable sources. This project uses Digital Image Correlation (DIC), as its main technique. DIC has the ability to record images of deformed surfaces, allowing for accurate analysis of the material being stressed under load. The strain distribution can then be obtained by applying the derivatives in the displacement field. This can be achieved by creating different methods like dots, grids, lines etc, on the specimen surface. The qualities of the results are mainly dependant on two factors: resolution of the camera, and quality of the speckle pattern on the specimen. Then, some image pre-processing will be applied on the captured images to get the desired output.

2.1.1 Different pattern used as reference entities of measurement

According to article published by Paulo Bastos de Castro (2012), the ability of digital image correlation to measure finite strains (whether localized or not) that result from inelastic deformations in metals or polymers has been demonstrated. Lines, grids, dots, and random arrays can all be used as reference entities of measurement when the DIC method is combined with a non-coherent light source.

SUTTON, M. A., 2009 said The use of a random pattern is one of the most popular strategies. Therefore, it is preferable that a specimen's surface texture exhibits suitable properties, such as a random distribution of grey intensity with discernible contrast. This condition may occur in nature on the specimen's surface (texture) or it may be created artificially, as in Figure 3, where the specimen was painted with an airbrush. The selection of a region of interest (ROI) where motion and/or deformation are anticipated to be observed is a crucial operational aspect of DIC. In this area, a virtual grid made of markers is established (see Figure 4).

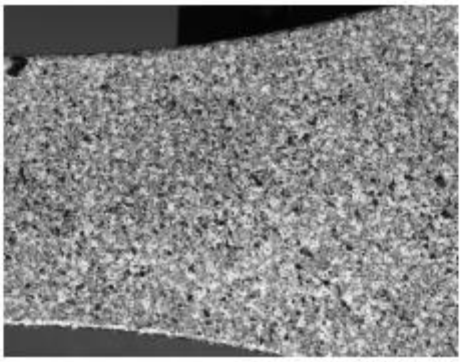


Figure 3: An example of a random speckle pattern.

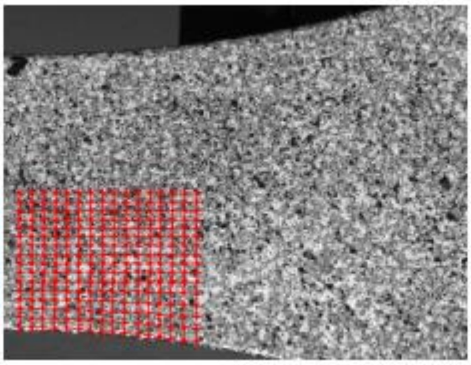


Figure 4: An example of an ROI and a grid of markers.

The DIC procedure, whose key features are described below, is how the movement of these markers is tracked across a series of images. Take a look at the ROI in both its pre- and post-deformed images. A subset of (2M + 1) (2M+1) pixels with the marker as its central pixel is defined for each marker of the grid in the undeformed image. This subset, which has its own distribution of grey intensities, will take up a different spot in the warped image.

2.2 Related Work/Previous Work

According to the journal published by Ahmad Fuad Ab Ghani (2018), DIC is an innovative technique which able to capture full field deformation of tensile deformation. The complex deformation captured for hybrid composite in-plane tensile deformation and behaviour using Digital Image Correlation (DIC) under static loading is a new area of study in literature.

Digital Image Correlation is a non-contact optical technology that captures digital photographs of an object's surface and then analyses the images to generate full-field deformation and measurements. This can be accomplished by drawing various methods on the specimen surface, such as dots, grids, lines, and so on. (Wei, K. S., Karuppanan, 2013).

According to Zhang Y, this technique begins with a reference image (before to loading), followed by a series of pictures obtained during the deformation. When compared to the initial non-distorted reference image, deformed images have a distinct dot pattern. These pattern differences can be estimated by correlating the pixels of the reference image with any warped image, and then computing a full-field displacement measurement. Applying the derivatives in the displacement field yields the strain distribution. To use this procedure, the surface of the object under examination must be prepped with a random dot pattern speckle pattern.

For surface deformation computation utilizing 2D Digital Image Correlation (DIC) technique, emphasized should be given on positioning of the specimen under testing, light intensity, and sources as well as the camera lens and its capability/resolution/frame rate of the camera. Accurate measurement relies heavily on imaging system configuration. In principle, a sample with random speckle pattern sprayed on the surface must be positioned perpendicular to the camera to avoid any out of plane motion. After the entire load applied events, a series of images are taken before and after loading and deformation and finally stored in the computer for post-processing images to obtain displacement contour/field using DIC algorithm. Basically from technical perspectives, for 2D DIC, image resolution plays a vital role in measurement accuracy. (Blaber, J., Adair, B., & Antoniou, A., 2015)

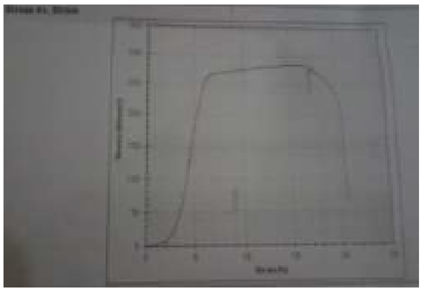
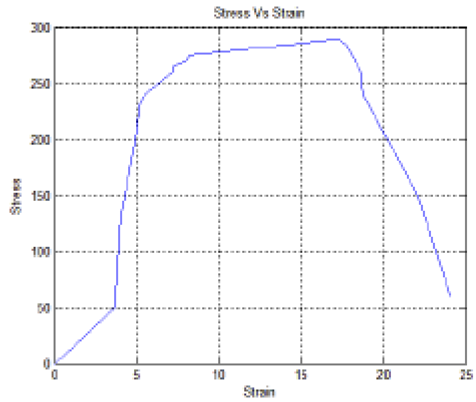
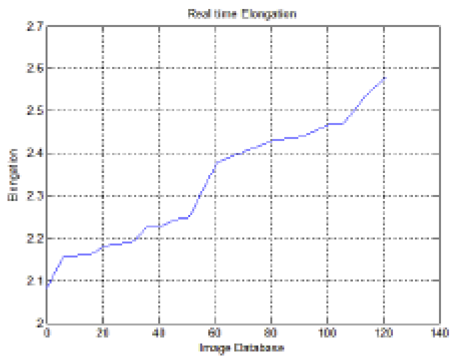
Miss Supriya S. Gadhe, 2016 said the qualities of the results are mainly dependant on two factors: resolution of the camera, and quality of the speckle pattern on the specimen. To obtain accurate results with the digital image correlation it is important to get an adequate speckle pattern. An adequate speckle pattern must have a considerable quantity of black speckles with different shapes and sizes. In order to get small and refine speckle pattern, air compressor and air brush kit will be used. The pattern will then be imaged under lighting level which maximized grey level distribution in the dynamic range.

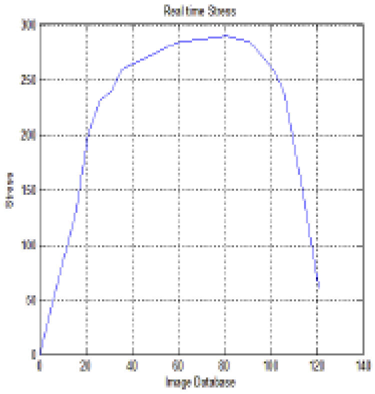
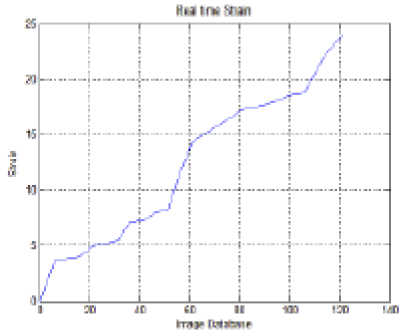
According to Neetu Rani (2017), a common technique for enhancing raw images that are obtained from different sources is image processing. It is a method for converting an image into digital form and applying specific actions to it in order to enhance the image or retrieve data from it. To produce images of higher quality, digital image processing employs a variety of methods, including correction, data formatting, and improved processes. Digital image processing basically consists of four operations: image pre-processing, image segmentation, feature extraction, and image classification.

Other studies on image processing can be seen in table 1 below.

Table 1: Review existing research paper

Title	Background	Framework	Result
The Novel Digital Image Correlation Technique in Predicting Behaviour and Failure of Hybrid Composite	This paper presents a technique in measuring deformation occurs on in-plane hybrid composite CFRP/GFRP. The challenging task of extracting mechanical properties of the hybrid composite is assisted with the use of Digital Image Correlation (DIC) technique. The main parameter that can be obtained during the tensile test is Ultimate Tensile Strength	• Sample Preparation • Formatting Displacements The **Ncorr** platform requires the user to input and define its displacement format which includes the measurement calibration and scale. These options were used to convert the displacements from pixels to real units. • Calculation of Strains • Plotting	- The modulus of elasticity in the longitudinal direction, *E1* for CFRP is around 120GPa - Modulus of elasticity for GFRP unidirectional 0 degree in accordance to ASTM D3039 is in the range of 40 to 45GPa. - The figure approaches 100GPa is explained by the fact that CFRP layers (3 layers as per layup) play a dominant factor in the stress-strain behavior of hybrid composite.  Figure 11: Strain contour for Ɛyy which average at 0.0018 for 4000N  Figure 12: Contour of strain, Ɛyy which average at 0.003 correspond to7000N

	(UTS), Yield Strength (σy), Elastic Modulus (E), Poisson Ratio (v), and percent elongation (ΔL)		
Digital Image Correlation Technique for Strain Measurement of Aluminium Plate	Digital Image Correlation (DIC) is an upcoming experimental stress –strain analysis technique and has certain advantages over others. DIC can be tested on almost all material with a large area of inspection and no pre-treatment is required as compared to other optical methods. DIC uses the principle of image correlation and produces strain field by analysing the movement of marked points on the subject. The Digital Image Correlation (DIC) is a state of art technique that can be used for accurate strain measurement. Because of its capability for	• Specimen The specimen whose strain is to be measured should have the random dot pattern. The random dot pattern is called as speckle pattern. The speckle pattern is necessary because it allows the software to identify and calculate the deformation with accuracy. • Digital Camera. The camera which we used for taking images is of sony xeperia T2 ultra. It is a high speed camera which has resolution of 16: 9 aspect ratio. • Digital Image Correlation Software i.e MATLAB MATLAB performs the image processing, correlation and finally we get the required output i.e. strain. Matlab is a high-level technical language. It has various number of inbuilt functions which are useful for image processing. Matlab can perform complex	The results are obtaining in the form of graphs by both tensile test done on UTM and by DIC method i.e by MATLAB  Fig8: Graph of strain versus stress obtain from UTM  Fig 9: Graph of strain versus stress obtain from DIC software  Fig 10: Graph of Elongation versus image database obtain from DIC software

	fast data acquisition, this technique is well suited for the characterization of material properties both in the elastic and plastic ranges. It also has advantages of full field, non-contact, and considerately high accuracy for displacement and strain measurements. The MATLAB software is an innovate system that uses the digital image correlation technique to provide strain measurements in a two-dimensional contour map for planar surface specimens.	image processing. Images can be added, subtracted or divided to detect the difference between two or more images.	 Fig11: Graph of stress versus image database obtain from DIC software  Fig 12: Graph of strain versus image database obtain from DIC software

The study found that both articles use different ways to find the strain from the image dataset. The first article use Ncorr to input and define its displacement format which includes the measurement calibration and scale. These options were used to convert the displacements from pixels to real units. While the second article use MATLAB software to perform the DIC technique. MATLAB performs the image processing, correlation and finally we get the required output i.e. strain.

In this project, we follow the framework of the second article because the expected output is to find the image correlation value.

2.3 Critical review of current problem and justification

After do some research, some of the methodology from other paper are quite related with this project title that make it suitable to be explored. The image preprocessing is the most important step before continue to the process or calculation of the analysis. The first preprocessing here is converting the image to the grayscale. Most of the paper about DIC will convert the image to grayscale to make it easier to find the strain measurement and detect the crack. The conversion to grayscale will help in order to find the material elongation.

The material used in recent research have the random dot pattern called speckle pattern. According to Miss Supriya S. Gadhe, 2016, the specimen should be sprayed with speckle pattern to get to have more accurate result. The speckle patterns must in black colour and of various shape and size. Speckles can be produced on surface specimen by using spray, brush or stamping.

There are several ways to calculate strain in this recent research. In *The Novel Digital Image Correlation Technique in Predicting Behavior and Failure of Hybrid Composite,* the calculation is done by looking at the output from Ncorr Post-Processing. It said the selection of the ideal strain radius is similar to the selection of the ideal subset radius, in that the smallest radius was desired which does not result in noisy strain data. The radius normally set to 13 because of the optimal radius. Meanwhile in *Digital Image Correlation Technique for Strain Measurement of Aluminium Plate*, there stated two methods to find strain, by taking tensile test on UTM and by using DIC technique. For tensile test by UTM machine, the specimen is fixed. The load is applied in KN starting from zero. The is load is applied in increasing manner till the specimen breaks and finally the graphs are plotted. For DIC method, the specimen is subjected to tensile test on UTM. The specimen is loaded till it breaks. The camera captures photos from when no load is applied to the specimen until the maximum load is applied to the specimen. As a result, the camera captures photographs of the specimens that are deforming when the force is applied at 3 second intervals. Specimen photos are taken until the specimen breaks. These images were exported to Matlab software for further correlation to achieve the objective.

The software and hardware that being used in other researches are same. Basically the hardware use is DIC machine to put a load on the material, Digital camera to capture multiple image or record the material that undergoing deformation and for software is MATLAB because there is an open source called Ncorr to verify the displacement of the control points for each specimen frame by frame. For this project, Matlab is used to do the preprocessing, calculation and a GUI manually without using Ncorr because the aim of this analysis is to find the elongation of the material.

2.4 Proposed Solution/further project

This paper describes how the strain from the DIC is found by obtaining it from the captured image. From the grayscale of the image, several calculations were applied until a strain was discovered. The methodology of this paper is referenced from other previous paper methodologies. This is because the desired result from this paper is the same as the previous paper, i.e. finding the stretch manually using Matlab.

2.5 Conclusion

In this chapter we already discussed about some of the previous works related to this topic and proposed solution for this project. We will see the details of the methodology for this project on the next chapter.

CHAPTER 3: PROJECT METHODOLOGY

3.1 Introduction

In this chapter, we will discuss about methodology that is being used for this project. Methodology is the details how the research was conducted, the research methods used and the reasons for choosing those methods.

3.2 Methodology

In this section, each element in the methodology will be discussed. These elements are as shown in Figure 5.

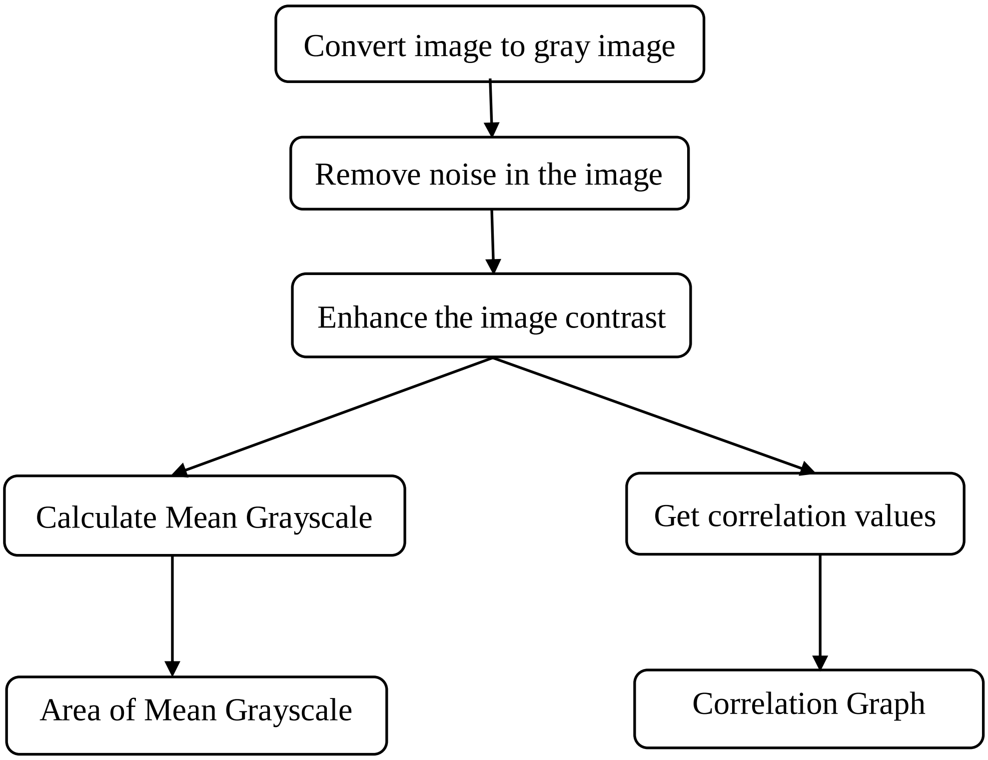


Figure 5: Flowchart of analysis DIC

Firstly, get the images input from the database and convert those images to gray image. It is the first image preprocessing that has been done. Next, remove all the noises at the images and then enhance the images contrast. These image preprocessing steps are to de-noise different types of noise and to improve the visibility of the image. After that, we divide it into two parts, first, calculate the gray scale mean to find the occurrence of elongation and use the area under the gray scale mean graph to prove the occurrence and the second is to get the correlation value to analyze the DIC technique of the image. Plot those values into a graph to make them easier to analyze.

3.2.1 Convert image to gray image

In digital images, grayscale means that the value of each pixel represents only the intensity information of the light. Such images typically display only the darkest black to the brightest white. To put it another way, the image only features black, white, and grey hues, with grey having many levels. The value of each pixel in a grayscale image is proportional to the number of bits of data utilized to represent it. The value of a grey image is commonly represented by 8 bits, that is, the pixel value of a pixel is represented by a combination of eight binary values. Therefore, the value range of pixels is 0–255, with a total of 256 grayscale levels.

A grayscale image is very helpful for further processing of segmentation. Here, all the analyses will be performed on the grayscale image. So, the input RGB fundus image (*I*) must be converted to a grayscale image (Ig) using equation (1) below.

$$I_g = 0.2989 \times IR + 0.5870 \times IG + 0.1140 \times IB \quad (1)$$

3.2.2 Noise removal

Image noise is an unavoidable side-effect occurring as a result of image capture, more simply understood as inaudible, yet inevitable fluctuations. In a digital camera, if the light which enters the lens misaligns with the sensors, it will create image noise. So here, median filter is applied to the image because it is easy to implement and it can be used for de-noising different types of noises. Figure 6 below shows the example for this process.

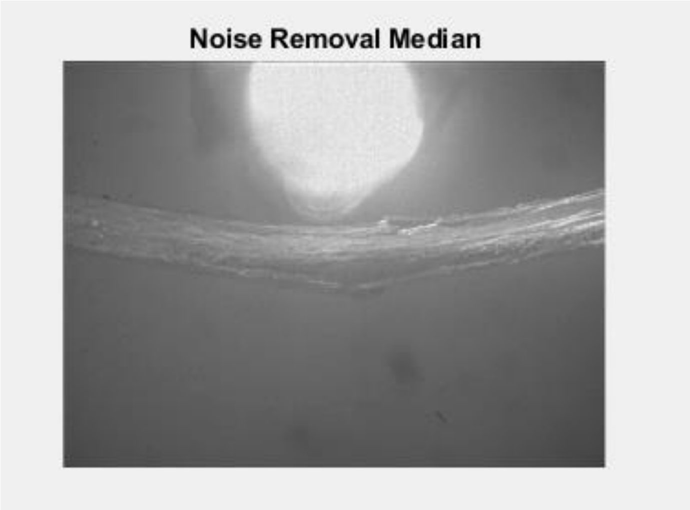


Figure 6: Example of Noise Removal

3.2.3 Contrast enhancement

Contrast enhancement processes adjust the relative brightness and darkness of objects in the scene to improve their visibility. The contrast and tone of the image can be changed by mapping the gray levels in the image to new values through a gray-level transform. It is a significant factor in any subjective evaluation of image quality which used to enhance the overall quality of the medical image for feature visualization and clinical measurement. Therefore, it is very important to enhance the contrast of such images before further processing and analysis. Below is Figure 7 shows that the image undergoes the process of contrast enhancement.

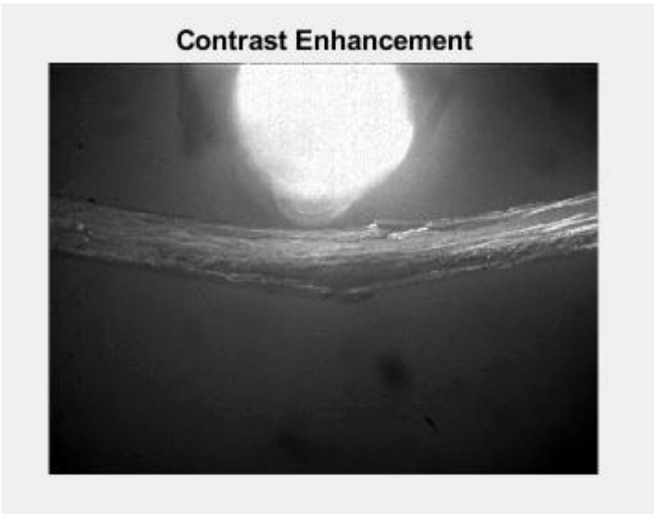


Figure 7: Example of Contrast Enhancement

3.2.4 Calculate Mean Grayscale

Elongation is a measure of deformation that occurs before a material eventually breaks when subjected to a tensile load. After the image is completed with the pre-processing stage in this analysis, the Region of Interest (ROI) needs to be set, shown in Figure 8. The most important thing to note here is that the ROI must be the same size arrangement for all images.

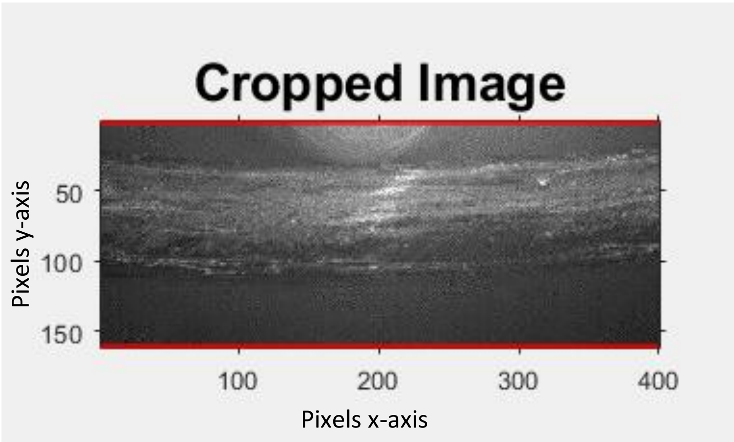


Figure 8: Cropped Image

Next, calculate the image coefficient. It can be referred in Figure 9. Since the grayscale has the value range of pixels 0-255, each scale contained in a given image will be divided by its sum to find the mean. Based on Figure 9, the mean was plotted with the blue line colour. The red colour line is the maximum values of each point and the green colour is a curve to fit the data based on the coefficients found. In Chapter 4.4, it will be explained more about this using the coding.

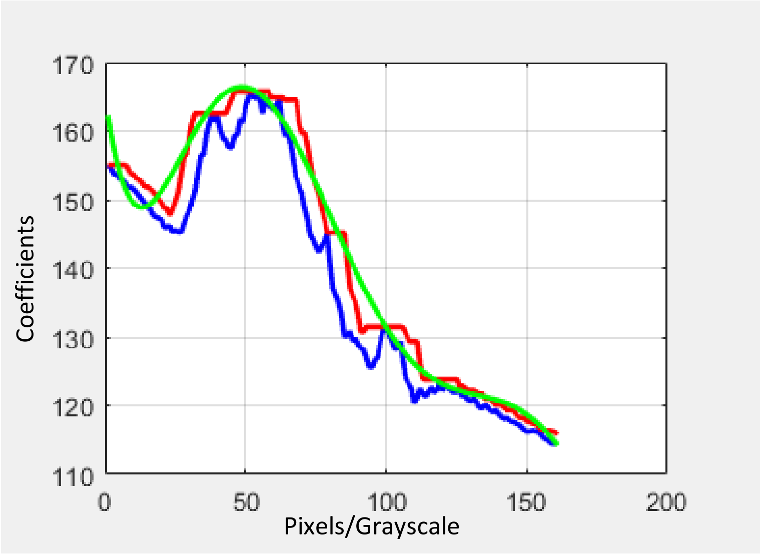


Figure 9: Coefficients

The width of threshold of the image is defined from the cropped image. The background corrected image is applied to identify the width as shown in Figure 10.

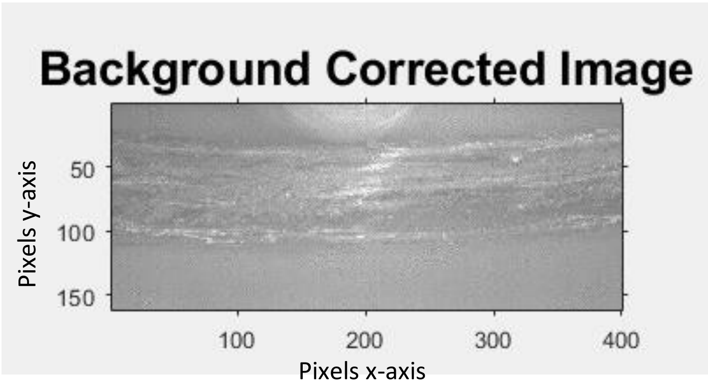


Figure 10: Background Corrected Image

Done this process for all image dataset until get the results as shown in Figure 11. Figure 11 shows a value of mean grayscale of background corrected image. x-axis indicates the image pixels while y-axis indicates the mean values. Get the values and plot all the points in one graph in the excel. Later, the elongation can be seen at Figure 13.

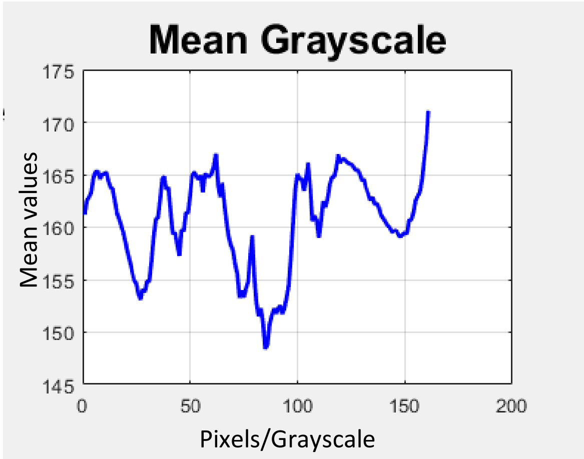


Figure 11: Graph of Width

3.2.5 Correlation Values

Graycoprops is a Gray-level co-occurrence matrix characteristic. Below is the syntax that will be used to find the correlation values.

```
stats = graycoprops(grayImage);
```

```
>> stats
stats =
  struct with fields:

       Contrast: 29645.9317599413
    Correlation: -0.00131975767997382
         Energy: 1.56527175662523e-05
    Homogeneity: 0.0250913040248958
```

Figure 12: How to find correlation values

In the Figure 12 above shows the values that we will get at the command window. Only the Correlation value that is used for all the images to be plotted into a graph later.

3.2.6 Area of Mean Grayscale

The result of this analysis is obtain in the form of graphs below.

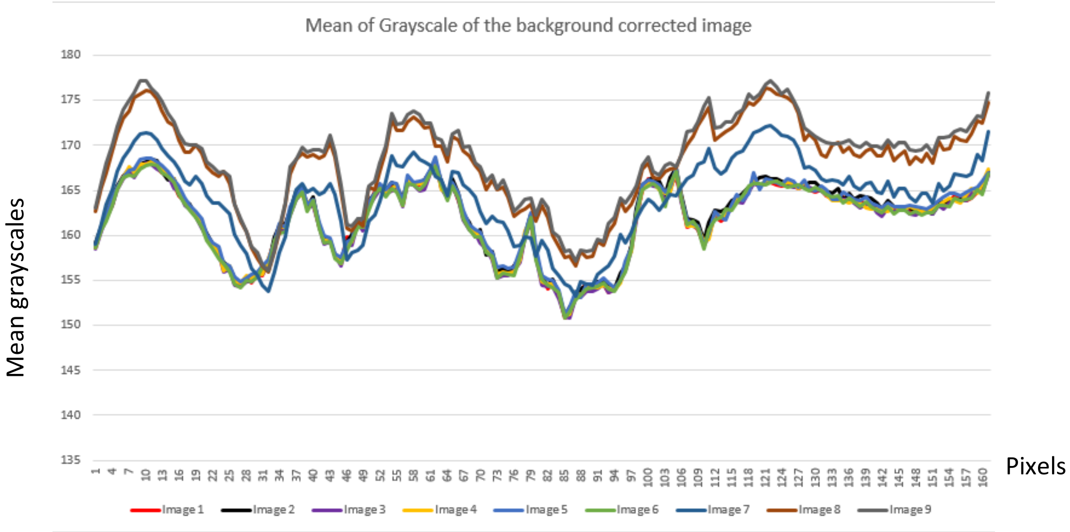


Figure 13: Graph of mean grayscale of the background corrected images

In Figure 13 above, it shows an increase in the mean value from one image to the next. All the mean values of the image are plotted in one graph using excel and it is quite difficult to read with the naked eye. So, the area under the graph of the mean grayscale in Figure 14 has proven it. It is also using excel. Image 1 is in the lowest position indicating that it has the lowest area under the graph followed by image 2, image 3, image 4, image 5, image 6, image 7, image 8, and image 9 at the top.

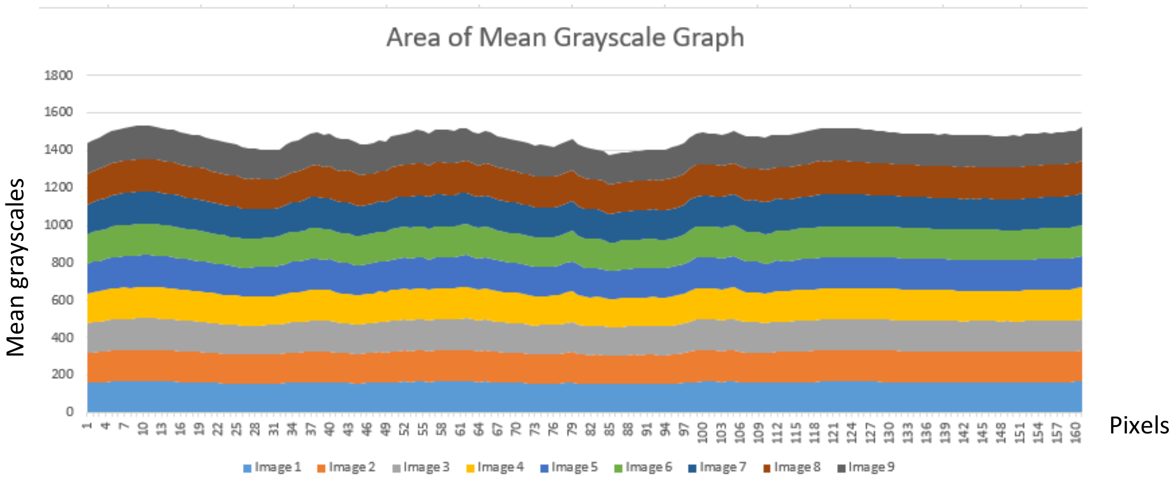


Figure 14: Area of Mean Grayscale Graph

Thus, it can be concluded that there is an occurrence of elongation through the above observation.

3.2.7 Correlation Graph

Below show the correlation values that already plotted into a graph.

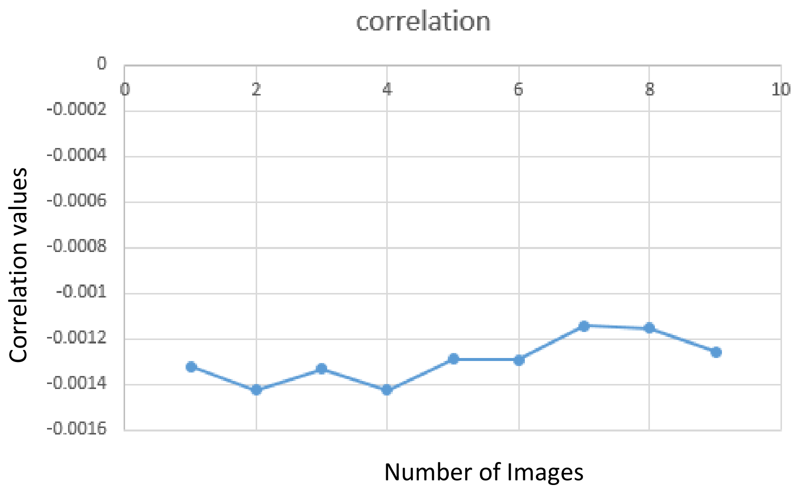


Figure 15: Correlation Graph

Figure 15 depicts the plots of correlation values that is get from all the images. We can see that the points are plotted at the negative values. Further explanation of this matter has been explained in Chapter 6.2.

3.3 Project Milestones

Table 2 depicts the project milestone of PSM 1 and Table 3 depicts the project milestone of PSM 2, which include the proposal phase through the project development phase and finally the project report phase.

Table 2: PSM 1

Assessment Name	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15
Proposal															
Project Progress 1															
Report Writing Progress 1															
Project Progress 2															
Report Writing Progress 2															
Demonstration (Supervisor)															
Report Evaluation (Supervisor)															
Demonstration (Evaluator)															
Report Evaluation (Evaluator)															
Presentation															

Table 3: PSM 2

Assessment Name	W1	W2	W3	W4	W5	W6	W7
Project Progress 1							
Project Progress 2							
Report Writing Progress							
Demonstration A (Supervisor)							
Demonstration B (Supervisor)							
Report Evaluation (Supervisor)							
English Proficiency							
Presentation							
Demonstration (Evaluator)							
Report Evaluation (Evaluator)							

3.4 Conclusion

In this chapter we already showed about the process and explain each of the process more details on how we identify the elongation. Some of the codes will be shown at the next chapter.

CHAPTER 4: DESIGN

4.1 Introduction

In this chapter defines the results of the analysis of the preliminary design and the result of the detailed design.

4.2 Network System Architecture

Basically, the system will have a Graphic User Interface (GUI) to select the DIC image from the file (database). After selecting image specimens, all images will go through an image preprocessing phase. A performance evaluation takes place and it will be plotted into a graph. Then, analyze the results. Figure 16 below is the data flow diagram that will be referred to.

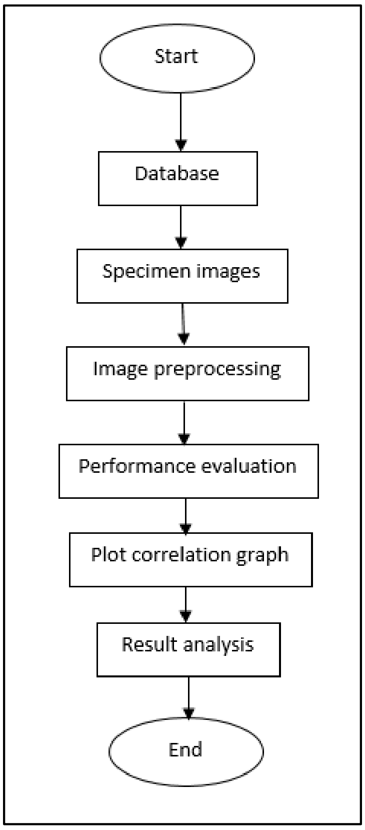


Figure 16: Data Flow Diagram of the system

4.3 Possible Scenarios

Most of the previous research paper just give the results of DIC in image form. There is not much research paper about DIC can be found easily and none of them has shown the results of the correlation coefficient of the DIC images. This is because DIC is a very new thing in the industrial. In this paper, it will help the other researcher in order to find the elongation or correlation coefficient of DIC image by insert the image into a GUI that will be created.

```
%% Read and convert to gray img
figure;
I = imread('s_8.jpg');
Igray = rgb2gray(I);
subplot(2,2,1), imshow(I), title('Original Image');
subplot(2,2,2), imshow(Igray),title('Greyscale');
```

Figure 17: Read and convert to Gray Image

```
%% Noise Removal
Kmedian = medfilt2(Igray);
subplot(2,2,3), imshow(Kmedian), title('Noise Removal Median');
```

Figure 18: Noise Removal

```
%% Contrast
Contrast = imadjust(Kmedian);
subplot(2,2,4), imshow(Contrast), title('Contrast Enhancement');
```

Figure 19: Contrast Enhancement

Figure 17, 18 and 19 show the code of converting the images to gray images, noise removal and contrast enhancement respectively that mentioned in Chapter 3. To explain Figures 17, 18 and 19 above, first, we need to read the desired image from the database, then use the ‘rgb2gray()’ function to convert the desired image from the database to a gray image. Show the image we got using ‘imshow()’ and name it Grayscale. The same goes for images called from the database by naming them as

Original Images so that it is easy to make comparisons between all images. Next, use the ‘medfilt2()’ function to remove the unnecessary noise at the Grayscale image and name it Noise Removal Median. Lastly, do the contrast enhancement to the Noise Removal Image by using ‘imadjust()’ function to enhance the image contrast and name it Contrast Enhancement. Figure 20 below shows the image results from those preprocessing techniques ie. Original Image, Grayscale, Noise Removal Median and Contrast Enhancement.

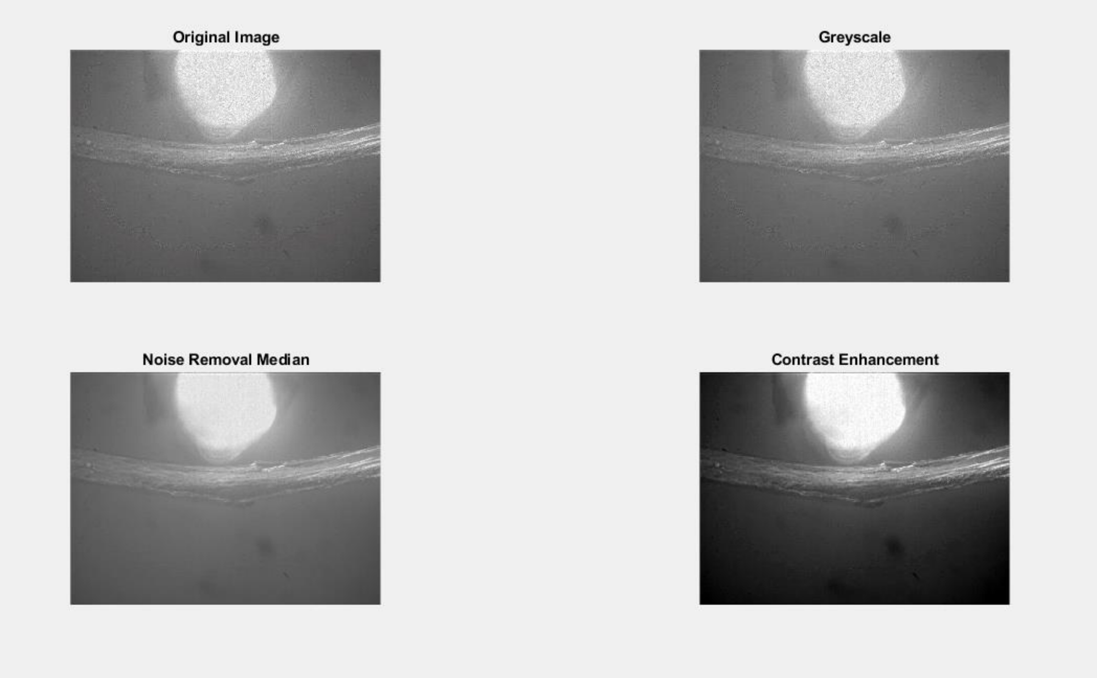


Figure 20: Preprocessing Image

Next, image is cropped to get the ROI. Figure 21 and Figure 22 below show on how to get ROI and display it manually using code.

```
col1 = 200;
col2 = 600;
rows1 = 210;
rows2 = 370;
xline(col1, 'Color', 'r', 'LineWidth', 2)
xline(col2, 'Color', 'r', 'LineWidth', 2)
yline(rows1,'Color', 'r', 'LineWidth', 2)
yline(rows2,'Color', 'r', 'LineWidth', 2)
grayImage = grayImage(rows1:rows2, col1:col2);
```

Figure 21: How to crop the desired segment

From Figure 21 above, we set the columns and rows of the image to get the region of interest (ROI). The columns refer to x-axis (xline) and the rows refer to y-axis (xline).

```
% Display the image.
subplot(2, 3, 2);
imshow(grayImage, []);
impixelinfo;
axis('on', 'image');
title('Cropped Image', 'FontSize', fontSize, 'Interpreter', 'None');
```

Figure 22: Display cropped image

Figure 22 indicates the codes to display the cropped image. The 'impixelinfo' is used to provide information of pixels when we display the grayImage and then name it Cropped Image.

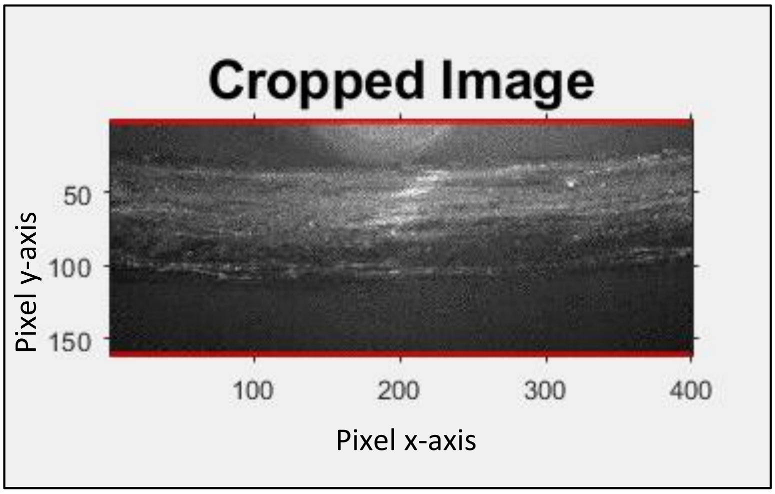


Figure 23: Result of displaying cropped image

The result from this process can be seen in Figure 23. We can see that it shows the pixel's information at the x-axis and y-axis of the image.

4.4 Metric Measurement

Here we will explain a bit about the calculation that already mentioned in Chapter 3. First, get the value of grayscale image and find the mean of it. Refer to Figure 24 for this process. The mean was plotted with the blue line colour. Then, find the maximum values of each point using ‘movmax’ and plot it with red colour. The ‘grid on’ and ‘hold on’ is to show the line grid on the graph and to hold the plot line together with the other plot line.

```
% Need to background correct so we can do a global threshold.
verticalProfile = mean(grayImage, 2);
subplot(2, 3, 3);
plot(verticalProfile, 'b-', 'LineWidth', 2);
grid on;
hold on;
verticalProfile = movmax(verticalProfile, 13);
plot(verticalProfile, 'r-', 'LineWidth', 2);
```

Figure 24: Calculate the coefficients of the image

In Figure 25, ‘Polyfit’ generates the coefficients of the polynomial, which can be used to model a curve to fit the data. The ‘Polyval’ evaluates a polynomial for a given set of x values. So, Polyval generates a curve to fit the data based on the coefficients found using polyfit.

```
% Fit to a quadratic.
xFit = 1 : length(verticalProfile);
coefficients = polyfit(xFit, verticalProfile, 6);
[rows, columns] = size(grayImage );
yFit = polyval(coefficients, xFit);
plot(yFit, 'g-', 'LineWidth', 2);
```

Figure 25: Plot the value to the graph

In Figure 26 below, we can see the result of mean vs grayscale graph which is the coefficient of the image. All the plot lines of the mean (blue), max (red), and polynomial (green) are plotted in one graph.

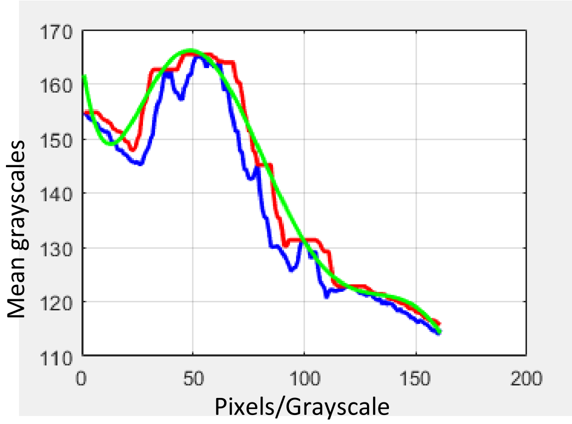


Figure 26: Graph of coefficients of the image

Figure 27 indicates how the background image is created. Background is a widely used approach to detect moving objects in a sequence of frames from static cameras. The base in this approach is that of detecting moving objects from the difference between the current frame and reference frame. So, to create the background image, we need to show the backgroundImage with the pixels that contain grayscale values from 0 to 255. Name it as Background Image.

```
% Display the image.
subplot(2, 3, 4);
imshow(backgroundImage, [0, 255]);
impixelinfo;
axis('on', 'image');
title('Background Image', 'FontSize', fontSize, 'Interpreter', 'None');
```

Figure 27: Create a background image

Next, find the background corrected image. It is the process of digitally manipulating image data such that the image's projection precisely matches a specific projection surface or shape. The gray image will be overlaid with the background image in Figure 27 to get the background corrected image like shown in Figure 29.

Equation (2) in Figure 28 shows how to overlay the gray image and the background image. The output of the process called Background Corrected Image.

$$grayImage = unit8(\max(backgroundImage(:)) * double(grayImage)\ .\!/\ backgroundImage) \quad (2)$$

```
% Display the background corrected image.
subplot(2, 3, 5);
grayImage = uint8(max(backgroundImage(:)) * double(grayImage) ./ backgroundImage);
imshow(grayImage, [0, 255]);
impixelinfo;
axis('on', 'image');
title('Background Corrected Image', 'FontSize', fontSize, 'Interpreter', 'None');
```

Figure 28: Display the background corrected image

The result from this process can be seen in Figure 29. We can see that it shows the pixel's information at the x-axis and y-axis of the image.

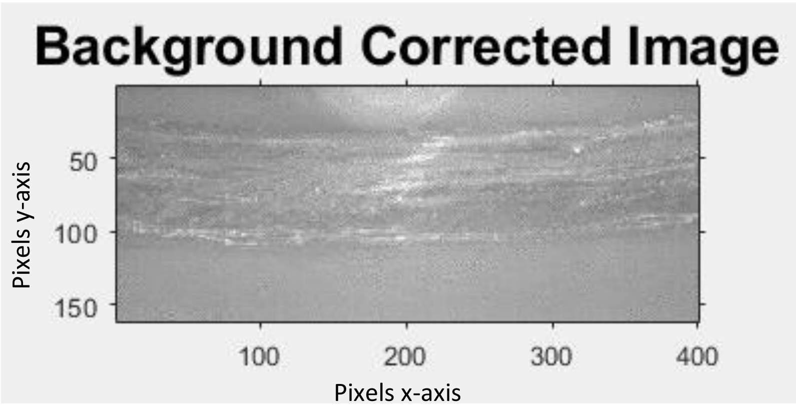


Figure 29: Results of background corrected image

In Figure 30, find the latest mean gray of the background corrected image using 'mean()' function and plot it in a graph with the blue line.

```
verticalProfile = mean(grayImage, 2);
subplot(2, 3, 6);
plot(verticalProfile, 'b-', 'LineWidth', 2);
grid on;
```

Figure 30: The Mean Grayscale codes

Figure 31 shows the mean gray image of the corrected image. We can see that it shows pixel values at x-axis and mean grayscale values at y-axis.

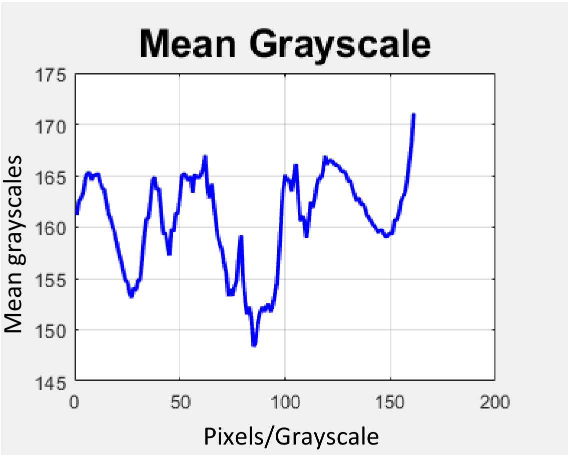


Figure 31: Graph of width

4.5 Conclusion

So, in this chapter all of the coding of the analysis is shown except for the GUI code. It will help a lot of researchers to find something new information about DIC since it is still new in this field and do not have a lot of research paper.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

In this chapter, the implementation between the processing codes and the GUI that was created using Matlab Software will be explained. In order to do so, App Designer is used which it lets user create professional apps in Matlab without having to be a professional software developer.

5.2 Environment Setup

The graphical user interface (GUI) can be designed using drag-and-drop, and it can quickly program its behavior using the integrated editor.

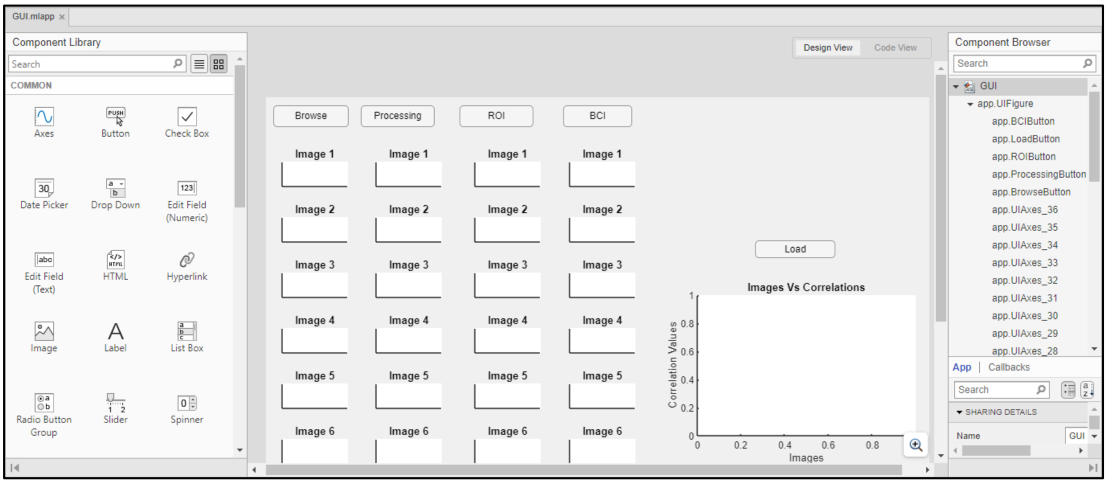


Figure 32: App Designer

Figure 32 contain the GUI of this project after the drag-and-drop of all the components needed. The most used components are the button and the axes. When the designing part was done, the button components and the axes components must be connected to each other. So, after the desired images is called, it can be processed by clicking at the button displayed only.

```
86          % Button pushed function: BrowseButton
87          function BrowseButtonPushed(app, event)
88              [filename, pathname] = uigetfile('*.*', 'Pick a Image');
89      if isequal(filename,0) || isequal(pathname,0)
90         helpdlg('User pressed cancel');
91      else
92      InputImage1=imread([pathname, filename]);
93      imshow(InputImage1, 'Parent',app.UIAxes);
94      app.InputImage1=InputImage1;
95
96      end
```

Figure 33: Code of Browse Button pushed function

In the component browser, click right at the button that want to start first and callback the button. It will be bringing to the Code View like shown in Figure 33. From line 88-93, it uses a same code from the Matlab to call the image from the existing file. Connect the button to the desired axes like shown in Figure 33 at line 93. Repeat the same code until it can call for all the desired images.

Do the same steps for all the buttons in the GUI like explained above. Figure 34, 35 and 36 show the code for Processing button, ROI button, and BCI button respectively. All the codes here are the same codes from the Matlab. Then, connect all those button to the desired axes like shown at line 191, line 296, and line 352. Repeat the same code until it can call for all the desired images.

```
179         % Button pushed function: ProcessingButton
180         function ProcessingButtonPushed(app, event)
181
182         InputImage1 = app.InputImage1;
183         [rows, columns, numberOfColorChannels] = size(InputImage1);
184         if numberOfColorChannels > 1
185             % It's not really gray scale like we expected - it's color.
186             % Extract the red channel (so the magenta lines will be white).
187             InputImage1 = InputImage1(:, :, 1);
188         end
189         Kmedian1 = medfilt2(InputImage1); %preprocessing
190         Contrast1 = imadjust(Kmedian1);
191         imshow(Contrast1,'Parent',app.UIAxes_10)
192         app.Contrast1=Contrast1;
```

Figure 34: Code of Processing Button pushed function

```
        % Button pushed function: ROIButton
        function ROIButtonPushed(app, event)
            Contrast1 = app.Contrast1;
        CropIm1 = imcrop(Contrast1,[200 210 400 160]); %[xmin ymin xlenght ylenght] | x=col y=row
        imshow(CropIm1,'Parent',app.UIAxes_19)
        app.CropIm1=CropIm1;
```

Figure 35: Code of ROI Button pushed function

```
        % Button pushed function: BCIButton
        function BCIButtonPushed(app, event)
            CropIm1=app.CropIm1;
            verticalProfile = mean(CropIm1, 2); % Mean of Matrix Row
            verticalProfile = movmax(verticalProfile, 13); %moving maximum
            xFit = 1 : length(verticalProfile);
            coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
            [rows, columns] = size(CropIm1);
            yFit = polyval(coefficients, xFit); %polyval
            backgroundImage = repmat(yFit(:), [1, columns]);
            B1 = uint8(max(backgroundImage(:)) * double(CropIm1) ./ backgroundImage);
            imshow(B1,'Parent',app.UIAxes_28);
```

Figure 36: Code of BCI Button pushed function

5.3 Conclusion

In conclusion, we used an App Designer to design the GUI and use some wanted codes from our previous codes on Matlab to be implemented in the App Designer.

CHAPTER 6: TESTING AND ANALYSIS

6.1 Introduction

In this phase, we will show the graph or the result of the graphical user interface (GUI) from what we have done in the implementation phase.

6.2 Results and Analysis

Figure 37 below shows the result after the images being processed, selected the ROI and Background Corrected Image (BCI). The BCI images will be used to get the values of the correlation. The values then will be plotted in the graph Image Vs Correlation like shown in Figure 39.

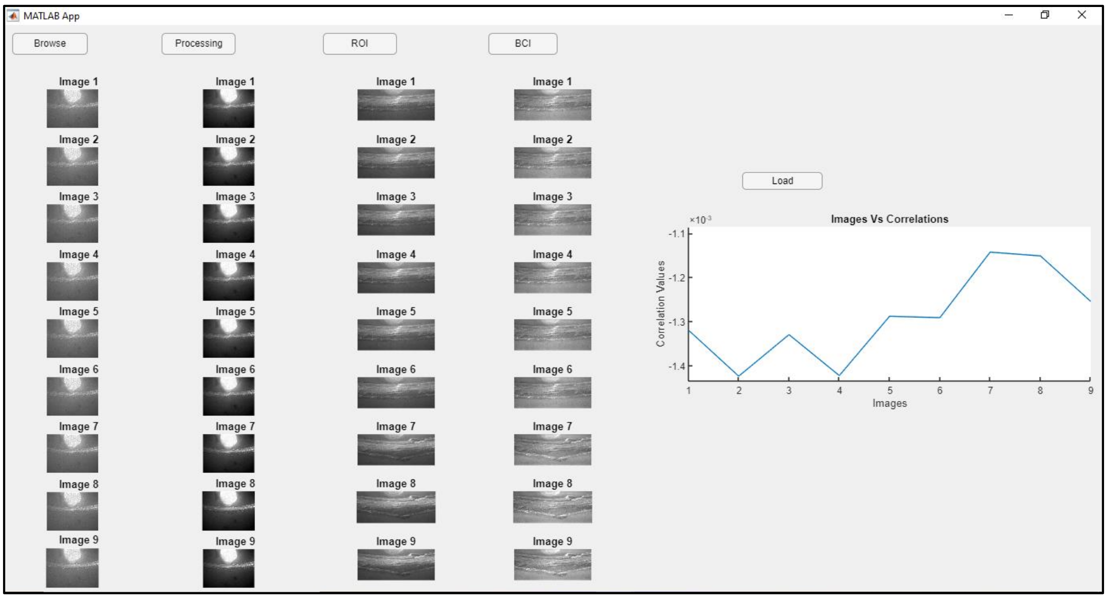


Figure 37: GUI of DIC technique using Image Preprocessing

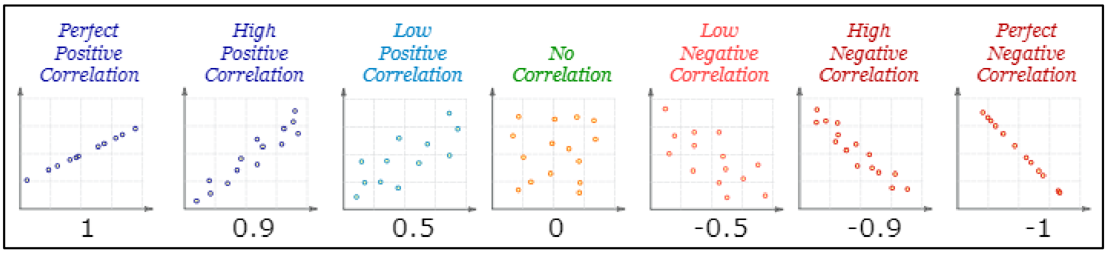


Figure 38: Correlation Graph

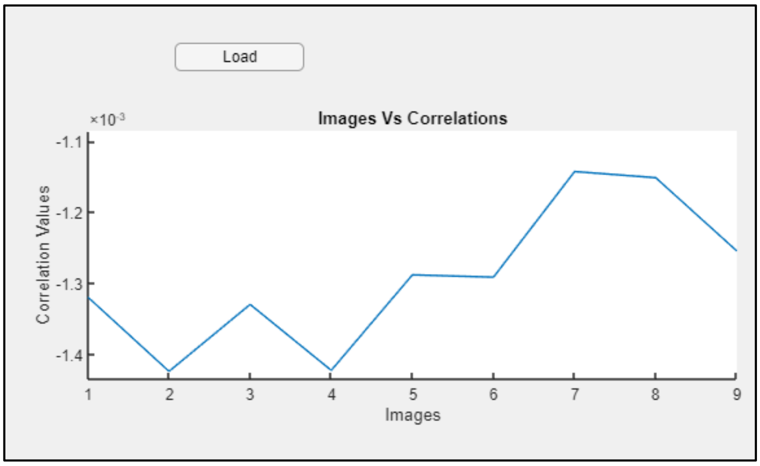


Figure 39: Images Vs Correlations

For your information, the values plotted in the graph is the value that we abstract from the excel file. This is one of the limitations of this system, where it still cannot find the correlation value directly from the image and continue to form it into a graph plot. So, from the Matlab command window, we find it manually using 'Graycoprops()' function (shown in Chapter 3.2.5), copy and paste the correlation values into excel named 'correlation'.

The value of correlations obtained from the images is in the negative like shown in Figure 39. The more the images are processed, the values increase and decrease until in the 7th image, the correlation value becomes the highest value. That value is the value closest to 0.

Referring to Figure 38, when the negative correlation value goes towards correlation 0, the molecules become spread. In other words, it can be said to be in a state of expansion. So, the correlation between the molecules is not so strong which can cause the specimen to break.

Figure 40: Zoom in to Image 7 where it starts to break

6.3 Conclusion

In this chapter, we analyze how the sought correlation value relates to the condition of the specimen on the given image dataset.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

Here is the last chapter in this report where we will conclude all about this project. There are also some limitations that will be mention below.

7.2 Project Summarization

Briefly, the system was built and designed using AppDesigner from Matlab to obtain correlation values using image processing. From those values, we can analyze why the specimen in the image dataset can break.

7.3 Project Contribution

The contribution from this DIC system is user can use this system to investigate the relationship between correlation and the image dataset. By looking at the correlation graph shown earlier, user can see how molecular diffusion can cause low intermolecular correlations. Therefore, the specimen will break easily.

7.4 Project Limitation

As mentioned above, there were several limitations that were detected throughout the project. The first one is cannot cancel or delete images once they are uploaded. If it is wrong to enter the image order, it is not possible to delete it as well.

Next, the number of image inputs are fixed which is 9. If there are more than the specified number, they will not be accepted and proceed to the next process. A low number of images can also cause the file image location to continue to appear and not be able to proceed to the next process.

7.5 Future Works

In the future, I hope this system can be upgraded by facing all the mentioned limitations and maybe can add more graphs or display the desired value obtained from the image dataset.

7.6 Conclusion

In conclusion, this DIC system works perfectly for those who want to explore another side of DIC rather than using NCORR. By referencing future work and limitations, the system still requires significant improvement. The goal outlined in Chapter 1 has been accomplished, and this project has progressed well within its intended parameters. I was able to investigate and evaluate the most recent research. I also successfully suggest an image processing-based framework for digital image correlation. Finally, I get to assess the accuracy of the analysis on the material that is under stress.

REFERENCES

CHAPTER 8: REFERENCES

[1] Rastogi, P. K., & Hack, E. (2012). Digital Image Correlation. In Optical methods for solid mechanics: A full-field approach (pp. 183–184). essay, Wiley-VCH.

[2] A. F. Ab Ghani, M. B. Ali, S. DharMalingam & J. Mahmud (2016). Digital Image Correlation (DIC) Technique in Measuring Strain Using Opensource Platform Ncorr. Journal of Advanced Research in Applied Mechanics, 10–21.

[3] Gustafson, H., Siegmund, G., & Cripton, P. (2016). Comparison of Strain Rosettes and Digital Image Correlation for Measuring Vertebral Body Strain. Journal of biomechanical engineering, 138(5),054501. https://doi.org/10.1115/1.4032799

[4] Ragupathy, P. (2013). (rep.). DIGITAL IMAGE CORRELATION APPLICATION IN TENSILE TEST FOR ALUMINIUM ACCORDING TO ASTM E-8 STANDARD (pp. 2–3). TRONOH, PERAK: UNIVERSITI TEKNOLOGI PETRONAS.

[5] Sutton, M. A., Orteu, J. J., and Schreier, H., (2009). Image correlation for shape, motion and deformation measurements: basic concepts, theory and applications. Springer US, New York.

[6] Hild, F., and Roux, S., (2012). Digital image correlation. Wiley-VCH, Weinheim, Germany.

[7] Gates, M., Lambros, J., and Heath, M. T. (2011). ―Towards High Performance Digital Volume Correlation‖. Experimental Mechanics, 51(4), Apr., pp. 491–507.

[8] Chu, T. C., Ranson, W. F., & Sutton, M. A. (1985). Applications of digital-image-correlation techniques to experimental mechanics. Experimental mechanics, 25(3), 232-244.

[9] Dong, Y. L., & Pan, B. (2017). A review of speckle pattern fabrication and assessment for digital image correlation. Experimental Mechanics, 57(8), 1161-1181.

[10] Mudassar, A. A., & Butt, S. (2016). Improved digital image correlation method. Optics and Lasers in Engineering, 87, 156-167.

[11] Pierron, F. & Sutton, M.A. (2010). Ultra high speed DIC and virtual fields method analysis of a three-point bending impact test on an aluminium bar. Experimental Mechanics 51(4): 537-563.

[12] Rani, Neetu. (2017). Image Processing Techniques: A Review. Journal on Today's Ideas - Tomorrow's Technologies. 5. 40-49. 10.15415/jotitt.2017.51003.

[13] de Castro, Paulo. (2012). Identification of constitutive parameters by using full-field measurements: an example of application to an elastoplastic damage model.

[14] Górszczyk, J., Malicki, K., & Zych, T. (2019). Application of Digital Image Correlation (DIC) Method for Road Material Testing. *Materials (Basel, Switzerland)*, *12*(15), 2349. https://doi.org/10.3390/ma12152349

APPENDIX

A) Source Codes

A: Preprocessing codes in MATLAB

```
%% Read and convert to gray img
%figure;
[fileName,pathName] = uigetfile('*.*');
I = importdata([pathName,fileName]);
%I = imread('q_1.jpg');
Igray = rgb2gray(I);
subplot(2,3,1), imshow(I), title('Original Image');
subplot(2,3,2), imshow(Igray),title('Greyscale');


%% Noise Removal
Kmedian = medfilt2(Igray);
subplot(2,3,3), imshow(Kmedian), title('Noise Removal Median');

%% Contrast
Contrast = imadjust(Kmedian);
subplot(2,3,4), imshow(Contrast), title('Contrast Enhancement');

%% Crop

CropIm = imcrop(Contrast,[200 210 400 160]);
subplot(2,3,5), imshow(CropIm), title('Crop Image');
subplot(2,3,6), imhist(CropIm);

```

B: Mean Grayscale to find elongation code in MATLAB

```
close all;  % Close all figures (except those of imtool.)
clear;  % Erase all existing variables. Or clearvars if you want.
figure;
workspace;  % Make sure the workspace panel is showing.
format long g;
format compact;
fontSize = 20;
markerSize = 40;
%--------------------------------------------------------------------
%    READ IN IMAGE
[fileName,pathName] = uigetfile('*.*');
grayImage = importdata([pathName,fileName]);
%grayImage = imread(Im);
% Get the dimensions of the image.
% numberOfColorChannels should be = 1 for a gray scale image, and 3 fc
[rows, columns, numberOfColorChannels] = size(grayImage);
if numberOfColorChannels > 1
    % It's not really gray scale like we expected - it's color.
    % Extract the red channel (so the magenta lines will be white).
    grayImage = grayImage(:, :, 1);
end
```

```
%----------------------------------------------------------------------
% Display the image.
subplot(2, 3, 1);
Kmedian = medfilt2(grayImage); %preprocessing
Contrast = imadjust(Kmedian);
imshow(Contrast, []);
impixelinfo;
axis('on', 'image');
title('Original Image', 'FontSize', fontSize, 'Interpreter', 'None');
hold on
drawnow;

% Maximize window.
g = gcf;
g.WindowState = 'maximized';
drawnow;

% Get rid of garbage in last two rows because the poster posted
% a screenshot instead of the actual image.
grayImage(end - 1, :) = grayImage(end - 2, :);
grayImage(end, :) = grayImage(end - 2, :);

```

```
%% Extract the middle
col1 = 200;
col2 = 600;
rows1 = 210;
rows2 = 370;
xline(col1, 'Color', 'r', 'LineWidth', 2)
xline(col2, 'Color', 'r', 'LineWidth', 2)
yline(rows1,'Color', 'r', 'LineWidth', 2)
yline(rows2,'Color', 'r', 'LineWidth', 2)
grayImage = grayImage(rows1:rows2, col1:col2);

% Display the image.
subplot(2, 3, 2);
imshow(grayImage, []);
impixelinfo;
axis('on', 'image');
title('Cropped Image', 'FontSize', fontSize, 'Interpreter', 'None');


```

```
%% Need to background correct so we can do a global threshold.
verticalProfile = mean(grayImage, 2); % Mean of Matrix Rows
subplot(2, 3, 3);
plot(verticalProfile, 'b-', 'LineWidth', 2);
grid on;
hold on;
verticalProfile = movmax(verticalProfile, 13); %moving maximum
plot(verticalProfile, 'r-', 'LineWidth', 2);

% Fit to a quadratic.
xFit = 1 : length(verticalProfile);
coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
[rows, columns] = size(grayImage );
yFit = polyval(coefficients, xFit); %polyval
plot(yFit, 'g-', 'LineWidth', 2);

%% Divide image by the background.
backgroundImage = repmat(yFit(:), [1, columns]);

% Display the image.
subplot(2, 3, 4);
imshow(backgroundImage, [0, 255]);
impixelinfo;
axis('on', 'image');
title('Background Image', 'FontSize', fontSize, 'Interpreter', 'None');
```

```
%% Display the background corrected image.
subplot(2, 3, 5);
grayImage = uint8(max(backgroundImage(:)) * double(grayImage) ./ backgroundImage);
imshow(grayImage, [0, 255]);
impixelinfo;
axis('on', 'image');
title('Background Corrected Image', 'FontSize', fontSize, 'Interpreter', 'None');
verticalProfile = mean(grayImage, 2); %mean of current gray image (background corrected image)
subplot(2, 3, 6);
plot(verticalProfile, 'b-', 'LineWidth', 2);
grid on;

stats = graycoprops(grayImage);
%% Find where the intensity falls below threshold.
threshold = 175;
row1 = find(verticalProfile < threshold, 1, 'first');
row2 = find(verticalProfile < threshold, 1, 'last');
width = row2 - row1;
caption = sprintf('Width = %d rows', width);
title(caption, 'FontSize', fontSize, 'Interpreter', 'None');

% Draw on image #2
subplot(2, 3, 2);
yline(row1, 'Color', 'r', 'LineWidth', 2)
yline(row2, 'Color', 'r', 'LineWidth', 2)
```

C: GUI code using AppDesigner

```
classdef GUI < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure           matlab.ui.Figure
        BCIButton          matlab.ui.control.Button
        LoadButton         matlab.ui.control.Button
        ROIButton          matlab.ui.control.Button
        ProcessingButton   matlab.ui.control.Button
        BrowseButton       matlab.ui.control.Button
        UIAxes_36          matlab.ui.control.UIAxes
        UIAxes_35          matlab.ui.control.UIAxes
        UIAxes_34          matlab.ui.control.UIAxes
        UIAxes_33          matlab.ui.control.UIAxes
        UIAxes_32          matlab.ui.control.UIAxes
        UIAxes_31          matlab.ui.control.UIAxes
        UIAxes_30          matlab.ui.control.UIAxes
        UIAxes_29          matlab.ui.control.UIAxes
        UIAxes_28          matlab.ui.control.UIAxes
        UIAxes3            matlab.ui.control.UIAxes
        UIAxes_27          matlab.ui.control.UIAxes
        UIAxes_26          matlab.ui.control.UIAxes
        UIAxes_25          matlab.ui.control.UIAxes
        UIAxes_24          matlab.ui.control.UIAxes
        UIAxes_23          matlab.ui.control.UIAxes
        UIAxes_22          matlab.ui.control.UIAxes
        UIAxes_21          matlab.ui.control.UIAxes
```

```
            UIAxes_20         matlab.ui.control.UIAxes
            UIAxes_19         matlab.ui.control.UIAxes
            UIAxes_18         matlab.ui.control.UIAxes
            UIAxes_17         matlab.ui.control.UIAxes
            UIAxes_16         matlab.ui.control.UIAxes
            UIAxes_15         matlab.ui.control.UIAxes
            UIAxes_14         matlab.ui.control.UIAxes
            UIAxes_13         matlab.ui.control.UIAxes
            UIAxes_12         matlab.ui.control.UIAxes
            UIAxes_11         matlab.ui.control.UIAxes
            UIAxes_10         matlab.ui.control.UIAxes
            UIAxes_9          matlab.ui.control.UIAxes
            UIAxes_8          matlab.ui.control.UIAxes
            UIAxes_7          matlab.ui.control.UIAxes
            UIAxes_6          matlab.ui.control.UIAxes
            UIAxes_5          matlab.ui.control.UIAxes
            UIAxes_4          matlab.ui.control.UIAxes
            UIAxes_3          matlab.ui.control.UIAxes
            UIAxes_2          matlab.ui.control.UIAxes
            UIAxes            matlab.ui.control.UIAxes
        end
```

```
        properties (Access = private)
            Property % Description
            InputImage1;
            InputImage2;
            InputImage3;
            InputImage4;
            InputImage5;
            InputImage6;
            InputImage7;
            InputImage8;
            InputImage9;
            Contrast1;
            Contrast2;
            Contrast3;
            Contrast4;
            Contrast5;
            Contrast6;
            Contrast7;
            Contrast8;
            Contrast9;
            CropIm1;
            CropIm2;
            CropIm3;
            CropIm4;
            CropIm5;
            CropIm6;
            CropIm7;
            CropIm8;
```

```
        % Callbacks that handle component events
        methods (Access = private)

            % Button pushed function: BrowseButton
            function BrowseButtonPushed(app, event)
                [filename, pathname] = uigetfile('*.*', 'Pick a Image');
        if isequal(filename,0) || isequal(pathname,0)
           helpdlg('User pressed cancel');
        else
        InputImage1=imread([pathname, filename]);
        imshow(InputImage1, 'Parent',app.UIAxes);
        app.InputImage1=InputImage1;

        end

        [filename, pathname] = uigetfile('*.*', 'Pick a Image');
        if isequal(filename,0) || isequal(pathname,0)
           helpdlg('User pressed cancel');
        else
        InputImage2=imread([pathname, filename]);
        imshow(InputImage2, 'Parent',app.UIAxes_2);
        app.InputImage2=InputImage2;

        end

        [filename, pathname] = uigetfile('*.*', 'Pick a Image');
        if isequal(filename,0) || isequal(pathname,0)
           helpdlg('User pressed cancel');
        else
        InputImage3=imread([pathname, filename]);
        imshow(InputImage3, 'Parent',app.UIAxes_3);
        app.InputImage3=InputImage3;

        end

        [filename, pathname] = uigetfile('*.*', 'Pick a Image');
        if isequal(filename,0) || isequal(pathname,0)
           helpdlg('User pressed cancel');
        else
        InputImage4=imread([pathname, filename]);
        imshow(InputImage4, 'Parent',app.UIAxes_4);
        app.InputImage4=InputImage4;

        end
```

```
        [filename, pathname] = uigetfile('*.*', 'Pick a Image');
        if isequal(filename,0) || isequal(pathname,0)
           helpdlg('User pressed cancel');
        else
        InputImage5=imread([pathname, filename]);
        imshow(InputImage5, 'Parent',app.UIAxes_5);
        app.InputImage5=InputImage5;

        end

        [filename, pathname] = uigetfile('*.*', 'Pick a Image');
        if isequal(filename,0) || isequal(pathname,0)
           helpdlg('User pressed cancel');
        else
        InputImage6=imread([pathname, filename]);
        imshow(InputImage6, 'Parent',app.UIAxes_6);
        app.InputImage6=InputImage6;

        end

        [filename, pathname] = uigetfile('*.*', 'Pick a Image');
        if isequal(filename,0) || isequal(pathname,0)
           helpdlg('User pressed cancel');
        else
        InputImage7=imread([pathname, filename]);
        imshow(InputImage7, 'Parent',app.UIAxes_7);
        app.InputImage7=InputImage7;
```

```

        end

        [filename, pathname] = uigetfile('*.*', 'Pick a Image');
        if isequal(filename,0) || isequal(pathname,0)
           helpdlg('User pressed cancel');
        else
        InputImage8=imread([pathname, filename]);
        imshow(InputImage8, 'Parent',app.UIAxes_8);
        app.InputImage8=InputImage8;

        end

        [filename, pathname] = uigetfile('*.*', 'Pick a Image');
        if isequal(filename,0) || isequal(pathname,0)
           helpdlg('User pressed cancel');
        else
        InputImage9=imread([pathname, filename]);
        imshow(InputImage9, 'Parent',app.UIAxes_9);
        app.InputImage9=InputImage9;

        end
            end
```

```
        % Button pushed function: ProcessingButton
        function ProcessingButtonPushed(app, event)

            InputImage1 = app.InputImage1;
            [rows, columns, numberOfColorChannels] = size(InputImage1);
            if numberOfColorChannels > 1
                % It's not really gray scale like we expected - it's color.
                % Extract the red channel (so the magenta lines will be white).
                InputImage1 = InputImage1(:, :, 1);
            end
            Kmedian1 = medfilt2(InputImage1); %preprocessing
            Contrast1 = imadjust(Kmedian1);
            imshow(Contrast1,'Parent',app.UIAxes_10)
            app.Contrast1=Contrast1;

            InputImage2 = app.InputImage2;
            [rows, columns, numberOfColorChannels] = size(InputImage2);
            if numberOfColorChannels > 1
                % It's not really gray scale like we expected - it's color.
                % Extract the red channel (so the magenta lines will be white).
                InputImage2 = InputImage2(:, :, 1);
            end
            Kmedian2 = medfilt2(InputImage2); %preprocessing
            Contrast2 = imadjust(Kmedian2);
            imshow(Contrast2,'Parent',app.UIAxes_11)
            app.Contrast2=Contrast2;

            InputImage3 = app.InputImage3;
            [rows, columns, numberOfColorChannels] = size(InputImage3);
            if numberOfColorChannels > 1
                % It's not really gray scale like we expected - it's color.
                % Extract the red channel (so the magenta lines will be white).
                InputImage3 = InputImage3(:, :, 1);
            end
            Kmedian3 = medfilt2(InputImage3); %preprocessing
            Contrast3 = imadjust(Kmedian3);
            imshow(Contrast3,'Parent',app.UIAxes_12)
            app.Contrast3=Contrast3;

            InputImage4 = app.InputImage4;
            [rows, columns, numberOfColorChannels] = size(InputImage4);
            if numberOfColorChannels > 1
                % It's not really gray scale like we expected - it's color.
                % Extract the red channel (so the magenta lines will be white).
                InputImage4 = InputImage4(:, :, 1);
            end
            Kmedian4 = medfilt2(InputImage4); %preprocessing
            Contrast4 = imadjust(Kmedian4);
            imshow(Contrast4,'Parent',app.UIAxes_13)
            app.Contrast4=Contrast4;

            InputImage5 = app.InputImage5;
            [rows, columns, numberOfColorChannels] = size(InputImage5);
            if numberOfColorChannels > 1
```

```
            % It's not really gray scale like we expected - it's color.
            % Extract the red channel (so the magenta lines will be white).
            InputImage5 = InputImage5(:, :, 1);
        end
        Kmedian5 = medfilt2(InputImage5); %preprocessing
        Contrast5 = imadjust(Kmedian5);
        imshow(Contrast5,'Parent',app.UIAxes_14)
        app.Contrast5=Contrast5;

        InputImage6 = app.InputImage6;
        [rows, columns, numberOfColorChannels] = size(InputImage6);
        if numberOfColorChannels > 1
            % It's not really gray scale like we expected - it's color.
            % Extract the red channel (so the magenta lines will be white).
            InputImage6 = InputImage6(:, :, 1);
        end
        Kmedian6 = medfilt2(InputImage6); %preprocessing
        Contrast6 = imadjust(Kmedian6);
        imshow(Contrast6,'Parent',app.UIAxes_15)
        app.Contrast6=Contrast6;

        InputImage7 = app.InputImage7;
        [rows, columns, numberOfColorChannels] = size(InputImage7);
        if numberOfColorChannels > 1
            % It's not really gray scale like we expected - it's color.
            % Extract the red channel (so the magenta lines will be white).
            InputImage7 = InputImage7(:, :, 1);
        end
```

```
        Kmedian7 = medfilt2(InputImage7); %preprocessing
        Contrast7 = imadjust(Kmedian7);
        imshow(Contrast7,'Parent',app.UIAxes_16)
        app.Contrast7=Contrast7;

        InputImage8 = app.InputImage8;
        [rows, columns, numberOfColorChannels] = size(InputImage8);
        if numberOfColorChannels > 1
            % It's not really gray scale like we expected - it's color.
            % Extract the red channel (so the magenta lines will be white).
            InputImage8 = InputImage8(:, :, 1);
        end
        Kmedian8 = medfilt2(InputImage8); %preprocessing
        Contrast8 = imadjust(Kmedian8);
        imshow(Contrast8,'Parent',app.UIAxes_17)
        app.Contrast8=Contrast8;

        InputImage9 = app.InputImage9;
        [rows, columns, numberOfColorChannels] = size(InputImage9);
        if numberOfColorChannels > 1
            % It's not really gray scale like we expected - it's color.
            % Extract the red channel (so the magenta lines will be white).
            InputImage9 = InputImage9(:, :, 1);
        end
        Kmedian9 = medfilt2(InputImage9); %preprocessing
        Contrast9 = imadjust(Kmedian9);
        imshow(Contrast9,'Parent',app.UIAxes_18)
        app.Contrast9=Contrast9;
```

```
        % Button pushed function: ROIButton
        function ROIButtonPushed(app, event)
            Contrast1 = app.Contrast1;
        CropIm1 = imcrop(Contrast1,[200 210 400 160]); %[xmin ymi
        imshow(CropIm1,'Parent',app.UIAxes_19)
        app.CropIm1=CropIm1;

        Contrast2 = app.Contrast2;
        CropIm2 = imcrop(Contrast2,[200 210 400 160]); %[xmin ymi
        imshow(CropIm2,'Parent',app.UIAxes_20)
        app.CropIm2=CropIm2;

        Contrast3 = app.Contrast3;
        CropIm3 = imcrop(Contrast3,[200 210 400 160]); %[xmin ymi
        imshow(CropIm3,'Parent',app.UIAxes_21)
        app.CropIm3=CropIm3;

        Contrast4 = app.Contrast4;
        CropIm4 = imcrop(Contrast4,[200 210 400 160]); %[xmin ymi
        imshow(CropIm4,'Parent',app.UIAxes_22)
        app.CropIm4=CropIm4;

        Contrast5 = app.Contrast5;
        CropIm5 = imcrop(Contrast5,[200 210 400 160]); %[xmin ymi
        imshow(CropIm5,'Parent',app.UIAxes_23)
        app.CropIm5=CropIm5;


        Contrast6 = app.Contrast6;
        CropIm6 = imcrop(Contrast6,[200 210 400 160]); %[xm
        imshow(CropIm6,'Parent',app.UIAxes_24)
        app.CropIm6=CropIm6;

        Contrast7 = app.Contrast7;
        CropIm7 = imcrop(Contrast7,[200 210 400 160]); %[xm
        imshow(CropIm7,'Parent',app.UIAxes_25)
        app.CropIm7=CropIm7;

        Contrast8 = app.Contrast8;
        CropIm8 = imcrop(Contrast8,[200 210 400 160]); %[xm
        imshow(CropIm8,'Parent',app.UIAxes_26)
        app.CropIm8=CropIm8;

        Contrast9 = app.Contrast9;
        CropIm9 = imcrop(Contrast9,[200 210 400 160]); %[xm
        imshow(CropIm9,'Parent',app.UIAxes_27)
        app.CropIm9=CropIm9;

        end
```

```
        % Button pushed function: BCIButton
        function BCIButtonPushed(app, event)
            CropIm1=app.CropIm1;
            verticalProfile = mean(CropIm1, 2); % Mean of Matrix Row
            verticalProfile = movmax(verticalProfile, 13); %moving maximum
            xFit = 1 : length(verticalProfile);
            coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
            [rows, columns] = size(CropIm1);
            yFit = polyval(coefficients, xFit); %polyval
            backgroundImage = repmat(yFit(:), [1, columns]);
            B1 = uint8(max(backgroundImage(:)) * double(CropIm1) ./ backgroundImage);
            imshow(B1,'Parent',app.UIAxes_28);

            CropIm2=app.CropIm2;
            verticalProfile = mean(CropIm2, 2); % Mean of Matrix Row
            verticalProfile = movmax(verticalProfile, 13); %moving maximum
            xFit = 1 : length(verticalProfile);
            coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
            [rows, columns] = size(CropIm2);
            yFit = polyval(coefficients, xFit); %polyval
            backgroundImage = repmat(yFit(:), [1, columns]);
            B2 = uint8(max(backgroundImage(:)) * double(CropIm2) ./ backgroundImage);
            imshow(B2,'Parent',app.UIAxes_29);
```

```
            CropIm3=app.CropIm3;
            verticalProfile = mean(CropIm3, 2); % Mean of Matrix Row
            verticalProfile = movmax(verticalProfile, 13); %moving maximum
            xFit = 1 : length(verticalProfile);
            coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
            [rows, columns] = size(CropIm3);
            yFit = polyval(coefficients, xFit); %polyval
            backgroundImage = repmat(yFit(:), [1, columns]);
            B3 = uint8(max(backgroundImage(:)) * double(CropIm3) ./ backgroundImage);
            imshow(B3,'Parent',app.UIAxes_30);

            CropIm4=app.CropIm4;
            verticalProfile = mean(CropIm4, 2); % Mean of Matrix Row
            verticalProfile = movmax(verticalProfile, 13); %moving maximum
            xFit = 1 : length(verticalProfile);
            coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
            [rows, columns] = size(CropIm4);
            yFit = polyval(coefficients, xFit); %polyval
            backgroundImage = repmat(yFit(:), [1, columns]);
            B4 = uint8(max(backgroundImage(:)) * double(CropIm4) ./ backgroundImage);
            imshow(B4,'Parent',app.UIAxes_31);

            CropIm5=app.CropIm5;
            verticalProfile = mean(CropIm5, 2); % Mean of Matrix Row
            verticalProfile = movmax(verticalProfile, 13); %moving maximum
            xFit = 1 : length(verticalProfile);
            coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
```

```
            [rows, columns] = size(CropIm5);
            yFit = polyval(coefficients, xFit); %polyval
            backgroundImage = repmat(yFit(:), [1, columns]);
            B5 = uint8(max(backgroundImage(:)) * double(CropIm5) ./ backgroundImage);
            imshow(B5,'Parent',app.UIAxes_32);

            CropIm6=app.CropIm6;
            verticalProfile = mean(CropIm6, 2); % Mean of Matrix Row
            verticalProfile = movmax(verticalProfile, 13); %moving maximum
            xFit = 1 : length(verticalProfile);
            coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
            [rows, columns] = size(CropIm6);
            yFit = polyval(coefficients, xFit); %polyval
            backgroundImage = repmat(yFit(:), [1, columns]);
            B6 = uint8(max(backgroundImage(:)) * double(CropIm6) ./ backgroundImage);
            imshow(B6,'Parent',app.UIAxes_33);

            CropIm7=app.CropIm7;
            verticalProfile = mean(CropIm7, 2); % Mean of Matrix Row
            verticalProfile = movmax(verticalProfile, 13); %moving maximum
            xFit = 1 : length(verticalProfile);
            coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
            [rows, columns] = size(CropIm7);
            yFit = polyval(coefficients, xFit); %polyval
            backgroundImage = repmat(yFit(:), [1, columns]);
            B7 = uint8(max(backgroundImage(:)) * double(CropIm7) ./ backgroundImage);
            imshow(B7,'Parent',app.UIAxes_34);
```

```
            CropIm8=app.CropIm8;
            verticalProfile = mean(CropIm8, 2); % Mean of Matrix Row
            verticalProfile = movmax(verticalProfile, 13); %moving maximum
            xFit = 1 : length(verticalProfile);
            coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
            [rows, columns] = size(CropIm8);
            yFit = polyval(coefficients, xFit); %polyval
            backgroundImage = repmat(yFit(:), [1, columns]);
            B8 = uint8(max(backgroundImage(:)) * double(CropIm8) ./ backgroundImage);
            imshow(B8,'Parent',app.UIAxes_35);

            CropIm9=app.CropIm9;
            verticalProfile = mean(CropIm9, 2); % Mean of Matrix Row
            verticalProfile = movmax(verticalProfile, 13); %moving maximum
            xFit = 1 : length(verticalProfile);
            coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
            [rows, columns] = size(CropIm9);
            yFit = polyval(coefficients, xFit); %polyval
            backgroundImage = repmat(yFit(:), [1, columns]);
            B9 = uint8(max(backgroundImage(:)) * double(CropIm9) ./ backgroundImage);
            imshow(B9,'Parent',app.UIAxes_36);
        end
```

```
        % Button pushed function: LoadButton
        function LoadButtonPushed(app, event)
            C = xlsread('correlation.xlsx');
            plot(app.UIAxes3,C);
        end
    end
```

```
        % Component initialization
        methods (Access = private)

            % Create UIFigure and components
            function createComponents(app)

                % Create UIFigure and hide until all components are created
                app.UIFigure = uifigure('Visible', 'off');
                app.UIFigure.Position = [100 100 826 683];
                app.UIFigure.Name = 'MATLAB App';

                % Create UIAxes
                app.UIAxes = uiaxes(app.UIFigure);
                title(app.UIAxes, 'Image 1')
                app.UIAxes.XTick = [];
                app.UIAxes.YTick = [];
                app.UIAxes.Position = [1 553 107 70];

                % Create UIAxes_2
                app.UIAxes_2 = uiaxes(app.UIFigure);
                title(app.UIAxes_2, 'Image 2')
                app.UIAxes_2.XTick = [];
                app.UIAxes_2.YTick = [];
                app.UIAxes_2.Position = [1 484 107 70];
```

```
                % Create UIAxes_3
                app.UIAxes_3 = uiaxes(app.UIFigure);
                title(app.UIAxes_3, 'Image 3')
                app.UIAxes_3.XTick = [];
                app.UIAxes_3.YTick = [];
                app.UIAxes_3.Position = [1 415 107 70];

                % Create UIAxes_4
                app.UIAxes_4 = uiaxes(app.UIFigure);
                title(app.UIAxes_4, 'Image 4')
                app.UIAxes_4.XTick = [];
                app.UIAxes_4.YTick = [];
                app.UIAxes_4.Position = [1 346 107 70];

                % Create UIAxes_5
                app.UIAxes_5 = uiaxes(app.UIFigure);
                title(app.UIAxes_5, 'Image 5')
                app.UIAxes_5.XTick = [];
                app.UIAxes_5.YTick = [];
                app.UIAxes_5.Position = [1 277 107 70];
```

```
% Create UIAxes_6
app.UIAxes_6 = uiaxes(app.UIFigure);
title(app.UIAxes_6, 'Image 6')
app.UIAxes_6.XTick = [];
app.UIAxes_6.YTick = [];
app.UIAxes_6.Position = [1 208 107 70];

% Create UIAxes_7
app.UIAxes_7 = uiaxes(app.UIFigure);
title(app.UIAxes_7, 'Image 7')
app.UIAxes_7.XTick = [];
app.UIAxes_7.YTick = [];
app.UIAxes_7.Position = [1 139 107 70];

% Create UIAxes_8
app.UIAxes_8 = uiaxes(app.UIFigure);
title(app.UIAxes_8, 'Image 8')
app.UIAxes_8.XTick = [];
app.UIAxes_8.YTick = [];
app.UIAxes_8.Position = [1 70 107 70];

```

```
% Create UIAxes_9
app.UIAxes_9 = uiaxes(app.UIFigure);
title(app.UIAxes_9, 'Image 9')
app.UIAxes_9.XTick = [];
app.UIAxes_9.YTick = [];
app.UIAxes_9.Position = [1 1 107 70];

% Create UIAxes_10
app.UIAxes_10 = uiaxes(app.UIFigure);
title(app.UIAxes_10, 'Image 1')
app.UIAxes_10.XTick = [];
app.UIAxes_10.YTick = [];
app.UIAxes_10.Position = [118 553 107 70];

% Create UIAxes_11
app.UIAxes_11 = uiaxes(app.UIFigure);
title(app.UIAxes_11, 'Image 2')
app.UIAxes_11.XTick = [];
app.UIAxes_11.YTick = [];
app.UIAxes_11.Position = [118 484 107 70];

```

```
            % Create UIAxes_12
            app.UIAxes_12 = uiaxes(app.UIFigure);
            title(app.UIAxes_12, 'Image 3')
            app.UIAxes_12.XTick = [];
            app.UIAxes_12.YTick = [];
            app.UIAxes_12.Position = [118 415 107 70];

            % Create UIAxes_13
            app.UIAxes_13 = uiaxes(app.UIFigure);
            title(app.UIAxes_13, 'Image 4')
            app.UIAxes_13.XTick = [];
            app.UIAxes_13.YTick = [];
            app.UIAxes_13.Position = [118 346 107 70];

            % Create UIAxes_14
            app.UIAxes_14 = uiaxes(app.UIFigure);
            title(app.UIAxes_14, 'Image 5')
            app.UIAxes_14.XTick = [];
            app.UIAxes_14.YTick = [];
            app.UIAxes_14.Position = [118 277 107 70];

            % Create UIAxes_15
            app.UIAxes_15 = uiaxes(app.UIFigure);
            title(app.UIAxes_15, 'Image 6')
            app.UIAxes_15.XTick = [];
            app.UIAxes_15.YTick = [];
            app.UIAxes_15.Position = [118 208 107 70];
```

```
            % Create UIAxes_16
            app.UIAxes_16 = uiaxes(app.UIFigure);
            title(app.UIAxes_16, 'Image 7')
            app.UIAxes_16.XTick = [];
            app.UIAxes_16.YTick = [];
            app.UIAxes_16.Position = [118 139 107 70];

            % Create UIAxes_17
            app.UIAxes_17 = uiaxes(app.UIFigure);
            title(app.UIAxes_17, 'Image 8')
            app.UIAxes_17.XTick = [];
            app.UIAxes_17.YTick = [];
            app.UIAxes_17.Position = [118 70 107 70];

            % Create UIAxes_18
            app.UIAxes_18 = uiaxes(app.UIFigure);
            title(app.UIAxes_18, 'Image 9')
            app.UIAxes_18.XTick = [];
            app.UIAxes_18.YTick = [];
            app.UIAxes_18.Position = [118 1 107 70];

            % Create UIAxes_19
            app.UIAxes_19 = uiaxes(app.UIFigure);
            title(app.UIAxes_19, 'Image 1')
            app.UIAxes_19.XTick = [];
            app.UIAxes_19.YTick = [];
            app.UIAxes_19.Position = [241 553 107 70];
```

```
% Create UIAxes_20
app.UIAxes_20 = uiaxes(app.UIFigure);
title(app.UIAxes_20, 'Image 2')
app.UIAxes_20.XTick = [];
app.UIAxes_20.YTick = [];
app.UIAxes_20.Position = [241 484 107 70];

% Create UIAxes_21
app.UIAxes_21 = uiaxes(app.UIFigure);
title(app.UIAxes_21, 'Image 3')
app.UIAxes_21.XTick = [];
app.UIAxes_21.YTick = [];
app.UIAxes_21.Position = [241 415 107 70];

% Create UIAxes_22
app.UIAxes_22 = uiaxes(app.UIFigure);
title(app.UIAxes_22, 'Image 4')
app.UIAxes_22.XTick = [];
app.UIAxes_22.YTick = [];
app.UIAxes_22.Position = [241 346 107 70];

% Create UIAxes_23
app.UIAxes_23 = uiaxes(app.UIFigure);
title(app.UIAxes_23, 'Image 5')
app.UIAxes_23.XTick = [];
app.UIAxes_23.YTick = [];
app.UIAxes_23.Position = [241 277 107 70];
```

```
            % Create UIAxes_24
            app.UIAxes_24 = uiaxes(app.UIFigure);
            title(app.UIAxes_24, 'Image 6')
            app.UIAxes_24.XTick = [];
            app.UIAxes_24.YTick = [];
            app.UIAxes_24.Position = [241 208 107 70];

            % Create UIAxes_25
            app.UIAxes_25 = uiaxes(app.UIFigure);
            title(app.UIAxes_25, 'Image 7')
            app.UIAxes_25.XTick = [];
            app.UIAxes_25.YTick = [];
            app.UIAxes_25.Position = [241 139 107 70];

            % Create UIAxes_26
            app.UIAxes_26 = uiaxes(app.UIFigure);
            title(app.UIAxes_26, 'Image 8')
            app.UIAxes_26.XTick = [];
            app.UIAxes_26.YTick = [];
            app.UIAxes_26.Position = [241 70 107 70];

            % Create UIAxes_27
            app.UIAxes_27 = uiaxes(app.UIFigure);
            title(app.UIAxes_27, 'Image 9')
            app.UIAxes_27.XTick = [];
            app.UIAxes_27.YTick = [];
            app.UIAxes_27.Position = [241 1 107 70];
```

```
% Create UIAxes3
app.UIAxes3 = uiaxes(app.UIFigure);
title(app.UIAxes3, 'Images Vs Correlations')
xlabel(app.UIAxes3, 'Images')
ylabel(app.UIAxes3, 'Correlation Values')
app.UIAxes3.Position = [499 228 319 229];

% Create UIAxes_28
app.UIAxes_28 = uiaxes(app.UIFigure);
title(app.UIAxes_28, 'Image 1')
app.UIAxes_28.XTick = [];
app.UIAxes_28.YTick = [];
app.UIAxes_28.Position = [359 553 107 70];

% Create UIAxes_29
app.UIAxes_29 = uiaxes(app.UIFigure);
title(app.UIAxes_29, 'Image 2')
app.UIAxes_29.XTick = [];
app.UIAxes_29.YTick = [];
app.UIAxes_29.Position = [359 484 107 70];

% Create UIAxes_30
app.UIAxes_30 = uiaxes(app.UIFigure);
title(app.UIAxes_30, 'Image 3')
app.UIAxes_30.XTick = [];
app.UIAxes_30.YTick = [];
app.UIAxes_30.Position = [359 415 107 70];
```

```
            % Create UIAxes_31
            app.UIAxes_31 = uiaxes(app.UIFigure);
            title(app.UIAxes_31, 'Image 4')
            app.UIAxes_31.XTick = [];
            app.UIAxes_31.YTick = [];
            app.UIAxes_31.Position = [359 346 107 70];

            % Create UIAxes_32
            app.UIAxes_32 = uiaxes(app.UIFigure);
            title(app.UIAxes_32, 'Image 5')
            app.UIAxes_32.XTick = [];
            app.UIAxes_32.YTick = [];
            app.UIAxes_32.Position = [359 277 107 70];

            % Create UIAxes_33
            app.UIAxes_33 = uiaxes(app.UIFigure);
            title(app.UIAxes_33, 'Image 6')
            app.UIAxes_33.XTick = [];
            app.UIAxes_33.YTick = [];
            app.UIAxes_33.Position = [359 208 107 70];

            % Create UIAxes_34
            app.UIAxes_34 = uiaxes(app.UIFigure);
            title(app.UIAxes_34, 'Image 7')
            app.UIAxes_34.XTick = [];
            app.UIAxes_34.YTick = [];
            app.UIAxes_34.Position = [359 139 107 70];

            % Create UIAxes_35
            app.UIAxes_35 = uiaxes(app.UIFigure);
            title(app.UIAxes_35, 'Image 8')
            app.UIAxes_35.XTick = [];
            app.UIAxes_35.YTick = [];
            app.UIAxes_35.Position = [359 70 107 70];

            % Create UIAxes_36
            app.UIAxes_36 = uiaxes(app.UIFigure);
            title(app.UIAxes_36, 'Image 9')
            app.UIAxes_36.XTick = [];
            app.UIAxes_36.YTick = [];
            app.UIAxes_36.Position = [359 1 107 70];

            % Create BrowseButton
            app.BrowseButton = uibutton(app.UIFigure, 'push');
            app.BrowseButton.ButtonPushedFcn = createCallbackFcn(app, @BrowseButtonPushed, true)
            app.BrowseButton.Position = [10 647 94 27];
            app.BrowseButton.Text = 'Browse';

            % Create ProcessingButton
            app.ProcessingButton = uibutton(app.UIFigure, 'push');
            app.ProcessingButton.ButtonPushedFcn = createCallbackFcn(app, @ProcessingButtonPushe
            app.ProcessingButton.Position = [119 647 92 27];
            app.ProcessingButton.Text = 'Processing';
```

```
            % Create ROIButton
            app.ROIButton = uibutton(app.UIFigure, 'push');
            app.ROIButton.ButtonPushedFcn = createCallbackFcn(app, @ROIButtonPushed, true);
            app.ROIButton.Position = [242 647 92 27];
            app.ROIButton.Text = 'ROI';

            % Create LoadButton
            app.LoadButton = uibutton(app.UIFigure, 'push');
            app.LoadButton.ButtonPushedFcn = createCallbackFcn(app, @LoadButtonPushed, true);
            app.LoadButton.Position = [610 484 100 22];
            app.LoadButton.Text = 'Load';

            % Create BCIButton
            app.BCIButton = uibutton(app.UIFigure, 'push');
            app.BCIButton.ButtonPushedFcn = createCallbackFcn(app, @BCIButtonPushed, true);
            app.BCIButton.Position = [371 647 86 27];
            app.BCIButton.Text = 'BCI';

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end
```

```

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = GUI

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```

B) Analysis of Data

A: Analysis for Correlation Sample 1

correlation
-0.001319758
-0.001424278
-0.001330009
-0.001422878
-0.001288394
-0.001291622
-0.001142882
-0.001151555
-0.001254783

B: Analysis for Correlation Sample 2

correlation sample 2
-0.001450438
-0.001426734
-0.001461011
-0.001453524
-0.001308191
-0.001617556
-0.001617556
-0.001399319
-0.001399319

C: Analysis for Mean Grayscale

Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9
158.6633	159.167	158.544	158.613	158.913	158.566	159.037	162.601	163.102
160.9327	161.05	160.519	160.733	160.925	160.599	161.344	165.302	165.825
162.0623	162.514	161.905	162.416	162.272	161.636	163.451	166.898	168.03
163.3466	164.012	163.262	163.82	164.18	163.384	165.08	169.332	170.105
165.389	165.471	165.032	165.239	165.324	165.05	167.102	171.292	172.12

166.4988	166.499	166.342	166.416	166.476	166.466	168.511	172.985	174.015
167.0923	167.359	167.07	167.691	167.282	166.728	169.449	173.823	175.057
166.7332	167.264	166.439	166.89	167.329	166.581	170.406	175.337	175.89
167.4838	167.86	167.676	167.958	168.387	167.421	171.204	175.681	177.127
167.8304	168.11	168.007	167.963	168.628	167.698	171.389	176.11	177.167
168.187	168.394	167.918	168.212	168.594	167.85	171.224	175.975	176.377
167.5436	168.324	167.319	168.002	168.224	167.698	170.616	175.209	175.763
166.98	167.135	167.052	167.389	167.713	167.012	169.688	173.868	174.89
166.7706	166.145	166.651	166.516	167.077	166.506	168.815	172.596	173.596
165.616	166.312	165.175	166.262	166.117	165.935	168.232	172.214	172.613
164.3915	164.539	164.945	164.763	165.531	164.484	167.299	170.524	171.304
163.6908	163.89	163.384	163.711	164.157	163.401	166.057	169.237	170.182
162.9601	163.414	162.955	163.142	163.352	162.805	165.586	169.274	170.025
162.0798	162.095	162.172	162.284	162.638	161.925	166.486	170.037	170.07
160.98	161.372	160.833	161.137	161.853	160.895	165.688	169.127	169.599
159.8978	160.209	159.791	159.973	160.072	159.392	164.337	167.676	168.19
158.808	158.985	158.663	158.845	159.209	158.529	163.576	166.988	167.596
157.7107	157.978	157.88	158.097	158.686	157.446	163.546	166.566	166.953
156.7057	156.985	155.888	156.127	156.913	156.865	163.095	167.08	167.062
156.2968	156.514	156.254	156.653	156.549	155.873	162.369	165.726	166.539
155.1022	155.219	154.496	154.898	155.399	154.646	160.157	163.401	163.436
154.389	154.731	154.339	154.768	154.788	154.175	158.9	161.621	161.783
155.1945	155.302	154.903	155.484	155.264	154.983	157.988	160.347	160.426
154.7581	155.476	154.798	155.207	155.683	154.863	156.354	158.571	158.718
155.6085	155.601	155.638	155.743	155.995	155.324	155.496	157.374	158.095

155.5486	156.559	155.601	155.651	156.594	156.409	154.461	156.636	156.726
157.2494	156.703	156.778	157.409	157.227	156.115	153.721	155.913	156.037
159.3915	159.833	159.137	159.125	159.269	159.471	155.903	158.204	158.399
161.0823	161.015	160.476	160.883	161.247	160.294	158.067	160.399	160.721
161.0723	161.254	160.701	160.948	160.865	160.374	159.796	163.519	163.576
162.7805	162.771	162.304	162.853	163.224	162.307	162.835	166.8	167.691
164.6409	164.88	164.04	164.469	164.441	164.005	165.022	168.444	168.494
164.7481	165.392	164.683	164.416	165.329	164.845	165.681	169.102	169.8
163.5761	163.137	163.015	163.768	163.504	162.596	164.815	168.713	169.237
163.6284	164.282	163.332	163.731	164.017	163.89	165.274	168.993	169.449
161.0773	161.09	160.923	160.988	161.394	160.733	164.549	168.529	169.471
159.2868	159.514	159.052	159.486	159.915	159.19	164.868	168.788	169.187
159.4564	159.439	159.229	159.429	159.641	159.347	165.751	170.347	171.127
157.7706	157.873	157.781	157.878	157.965	157.344	164.499	168.758	169.237
156.9701	157.077	156.576	156.873	157.591	156.818	161.606	164.988	165.623
159.7731	159.324	158.993	158.985	159.247	158.362	157.324	160.721	161.18
159.5736	160.062	158.838	159.224	159.623	159.449	158.077	160.476	161.097
161.3815	161.721	161.424	161.913	161.803	160.928	158.302	161.546	161.87
161.3317	161.636	160.476	160.968	161.202	160.925	158.928	160.988	161.888
163.0948	163.82	163.237	163.312	163.748	162.723	161.608	165.327	165.466
164.9701	164.618	164.242	164.656	164.726	164.524	162.259	165.105	165.975
165.2269	165.434	165.102	165.424	165.763	165.157	163.791	167.06	168.02
165.0549	165.05	164.623	164.771	165.052	164.262	165.953	169.312	169.898
165.5436	165.873	164.975	165.107	165.905	165.244	168.803	172.89	173.521
165.3616	165.342	165.252	165.788	165.703	164.87	167.825	171.678	172.287

163.4214	164.17	163.155	163.334	163.691	163.357	167.648	171.686	172.459
165.8554	165.88	165.753	166.12	166.663	165.718	168.489	172.636	173.364
165.8055	165.843	165.668	165.783	165.858	165.618	169.217	173.14	173.845
165.6334	165.838	164.983	165.743	166.035	165.115	168.481	172.656	173.411
166.1072	166.102	165.107	165.748	166.16	165.636	168.165	171.883	172.414
166.9476	167.192	166.985	167.047	167.426	166.95	167.803	172.047	172.491
168.2045	168.09	168.277	168.584	168.751	167.661	166.601	169.958	170.838
165.3067	166.04	165.155	165.13	165.843	165.638	166.09	169.973	170.554
164.2968	164.077	164.095	164.723	164.658	163.818	164.446	168.192	168.995
165.798	166.232	165.524	165.471	165.82	165.479	167.112	171.05	171.322
163.9252	164.636	163.958	164.521	164.676	164.132	166.943	170.531	171.628
162.3516	162.319	161.768	162.392	162.748	161.848	165.618	169.344	169.783
160.9476	161.456	160.613	160.791	161.132	160.613	165.087	168.893	169.898
160.2145	159.855	160.264	160.334	160.616	159.858	164.309	167.726	168.08
159.788	160.696	159.195	159.873	160.372	159.875	162.691	167.025	167.676
158.5636	157.848	158.436	158.86	158.933	158.04	161.359	165.072	165.975
157.6733	158.19	157.214	157.549	158.132	157.666	162.065	166.137	166.716
155.4988	155.828	155.267	155.726	156.406	155.212	161.514	165.09	165.227
156.2394	156.524	155.566	155.938	156.608	155.756	161.411	165.364	166.162
155.6409	155.823	155.559	155.858	156.359	155.616	160.559	164.309	164.833
156.5461	156.539	156.032	155.925	156.499	155.566	158.728	162.14	162.599
157.2718	157.658	157.07	157.307	158.374	157.731	158.908	162.701	163.322
159.9401	160.042	159.638	160.499	160.082	159.608	159.83	162.948	163.953
161.7781	162.389	161.574	161.693	162.476	162.045	159.651	163.491	164.112
157.7431	158.062	157.676	158.07	158.773	157.247	157.656	161.526	162.227

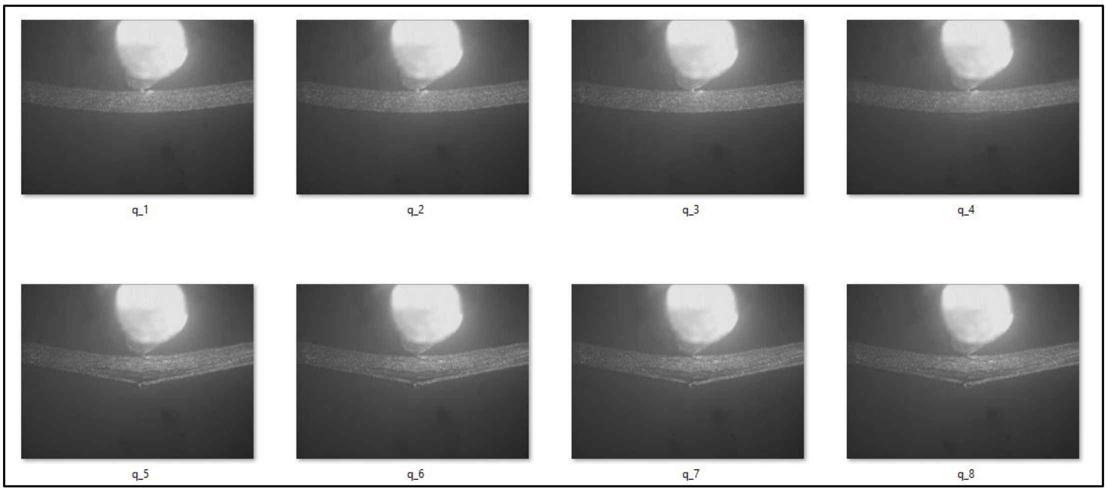
155.2594	155.357	154.489	154.808	155.474	155.107	159.456	163.329	163.98
153.99	154.491	154.342	154.536	155.085	154.441	158.334	162.032	163.032
154.7506	155.15	154.329	154.539	154.96	154.17	156.377	159.968	160.337
153.3716	153.539	153.035	153.579	153.945	153.643	155.549	158.796	159.865
150.808	151.137	151.005	151.15	151.232	150.873	154.549	157.536	158.167
151.1247	151.993	150.838	151.287	152.065	151.504	154.277	157.688	158.369
153.1721	153.002	153.037	153.83	153.773	152.813	153.177	156.589	157.18
153.8753	154.092	153.155	153.177	153.162	153.382	154.825	158.257	158.322
154.5212	154.449	153.761	154.219	154.546	154.09	154.524	157.586	158.222
153.98	154.549	153.838	154.364	154.591	154.145	154.444	157.628	158.329
154.3541	154.805	154.065	154.209	154.743	154.277	155.596	159.309	159.516
154.5586	154.855	154.603	154.409	155.277	154.671	156.185	158.925	159.317
153.6708	154.491	153.633	154.155	154.693	154.03	156.663	160.419	161.309
154.0474	154.09	153.925	153.853	154.135	153.756	157.623	161.334	161.973
155.1122	155.766	154.94	154.718	155.007	154.97	160.137	163.923	164.165
156.1546	156.658	156.11	156.808	157.152	155.97	159.17	162.623	163.561
158.7382	158.898	158.332	158.262	158.581	158.451	160.394	163.519	164.596
161.99	162.928	161.571	162.177	162.364	162.195	161.808	164.733	165.461
164.9551	165.678	165.309	165.087	165.641	165.092	163.067	167.424	167.766
166.2768	166.19	165.788	165.835	166.207	165.536	164.022	168	168.753
165.6584	166.459	165.501	165.828	166.037	165.766	163.541	166.409	167.112
165.4963	166.052	165.324	165.037	165.551	165.237	162.731	166.229	166.706
164.2469	164.539	163.9	164.618	164.808	163.15	163.746	167.085	167.796
165.3965	166.459	164.933	164.793	165.406	165.691	164.516	167.344	168.05
166.6185	167.491	167.195	167.411	167.549	167.1	164.399	167.549	167.579

164.0723	164.616	163.898	164.411	164.214	164.18	165.339	168.815	168.863
160.8404	161.9	161.16	161.002	161.768	161.207	166.349	170.09	171.551
161.1746	161.726	161.065	161.012	161.284	161.541	166.753	170.963	171.648
160.6633	161.421	160.756	160.9	161.065	160.761	167.883	172.142	172.631
158.8529	159.414	158.91	159.294	158.885	158.451	168.12	173.237	174.387
160.2668	161.434	159.868	159.559	160.267	160.608	169.706	174.197	175.229
162.1172	162.773	162.401	162.511	162.594	161.698	167.481	170.648	171.933
161.5187	162.691	161.933	162.132	161.98	162.04	166.788	171.182	172.055
162.0324	162.98	162.127	162.162	161.748	162.905	167.224	171.464	172.546
163.616	163.798	163.117	163.312	163.411	162.818	168.539	171.908	172.556
164.1172	164.195	163.791	164.07	164.539	163.606	169.107	172.434	173.486
164.1596	164.651	163.576	163.993	163.888	164.384	169.429	173.8	174.005
164.9352	165.349	164.865	164.885	164.648	165.519	170.434	174.805	175.681
166.2544	165.878	165.771	166.399	166.98	165.698	171.406	174.544	175.135
165.4264	166.382	165.466	165.409	165.075	165.733	171.641	175.214	175.753
165.8304	166.606	165.776	166.147	166.272	165.561	172.037	176.426	176.828
165.8529	166.327	166.055	166.05	166.192	165.84	172.252	176.262	177.147
165.6085	166.327	165.835	165.975	165.908	166.065	171.651	175.751	176.544
165.4663	165.965	165.763	165.574	165.576	165.733	171.085	175.541	175.731
165.6085	165.688	165.651	165.858	166.247	165.384	170.955	175.242	176.269
165.4738	165.686	165.304	165.731	166.035	165.436	170.536	174.696	175.08
165.2219	165.486	165.319	165.197	165.337	165.596	169.646	173.539	173.953
165.4214	165.347	165.481	166.017	166.137	165.262	167.449	170.576	171.89
165.1347	165.885	165.352	165.1	165.08	164.93	167.633	171.279	171.536
164.7681	165.915	165.105	165.122	164.908	165.352	167.185	170.653	171.027

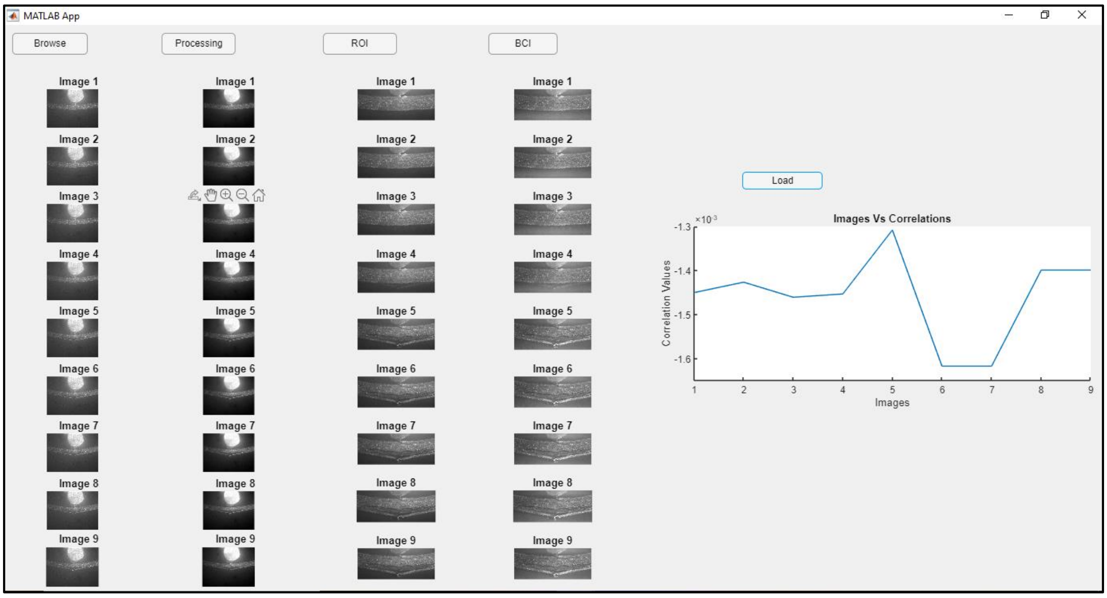
165.0998	165.269	164.928	165.297	165.456	164.91	166.234	168.95	170.589
164.4564	164.93	164.541	164.536	165.01	164.526	165.975	168.738	170.302
164.1222	164.673	164.294	163.868	164.017	164.204	166.155	170.022	170.175
163.798	165.15	163.985	163.845	163.983	164.377	166.037	170.279	170.227
164.2594	163.873	164.037	163.868	164.741	163.641	165.554	169.257	170.319
163.7955	164.686	163.958	163.556	164.372	164.152	166.603	169.781	170.554
164.0324	164.01	163.549	163.855	163.988	163.691	165.521	168.983	170.005
163.8628	164.446	163.092	163.544	163.828	163.224	165.057	168.681	169.825
163.404	164.322	163.421	163.092	164.272	163.671	165.778	169.337	170.282
163.2594	164.182	163.441	162.948	163.651	163.182	165.888	169.603	169.913
163.212	163.564	162.501	162.958	163.309	162.83	164.768	168.783	169.97
163.0748	162.965	162.167	162.723	163.06	162.551	164.723	168.81	169.643
163.0798	163.843	163.102	162.656	163.556	163.234	166.052	170.057	170.539
162.7656	163.145	162.601	163.165	163.162	162.369	164.324	168.299	169.723
163.0948	163.212	163.077	163.03	163.19	162.828	165.252	168.81	170.382
163.0773	163.207	163.065	162.97	163.185	162.778	165.224	169.439	170.302
162.8329	163.072	162.399	163.135	163.314	162.369	164.232	167.92	169.506
162.6085	162.646	162.187	162.945	163.155	162.676	163.781	168.641	169.352
162.8603	162.86	162.628	162.706	162.99	162.307	164.681	168.227	169.511
162.8803	162.601	162.726	162.616	162.88	162.541	164.696	169.047	170.05
162.7107	162.421	162.364	163.115	163.289	162.484	163.653	167.97	169.145
163.7631	163.658	163.354	163.272	163.9	163.294	165.688	170.431	170.798
163.4414	163.681	162.915	163.711	164.247	163.337	164.888	169.506	170.833
163.7307	163.708	163.574	164.127	164.658	163.234	165.414	169.616	170.988
164.3566	164.03	163.963	163.868	164.641	164.142	166.796	170.93	171.484

164.1397	164.045	163.818	163.536	164.464	163.868	166.701	170.628	171.736
163.818	164.077	163.89	164.514	164.763	163.985	166.579	170.404	171.509
164.1646	164.399	164.736	165.239	165.13	164.102	166.853	171.234	172.277
164.8529	165.137	165.02	164.658	165.347	165.065	168.948	172.8	173.332
165.2544	164.935	164.756	165.783	166.025	164.556	168.249	172.494	173.162
166.8554	167.15	166.673	167.329	167.017	166.863	171.556	174.803	175.8

C) Image Dataset Sample 2



D) Output for Image Dataset Sample 2



E) Plagiarism