ANALYSIS ON DIGITAL IMAGE CORRELATION USING IMAGE PROCESSING

A PROJECT REPORT

ABSTRACT

A visual object identification method called Digital Image Correlation (DIC) may be utilized to quantify object deformation. This method allows for the correlation of digital photographs of an item before and after deformation as well as the calculation of an object's deformation and strain field using a point on the image. This method is typically used in engineering with MATLAB. The discovery of strain is not what we want to find in this paper, but rather analysis of the DIC technique using image processing because there is little research on image processing being applied to the DIC technique, making it difficult for other researchers to do analysis on it. So, the significance of this project is that by implementing the proposed algorithm in their application, developers can greatly benefit the community by providing information about the DIC. This project is also carried out with the help of the MATLAB software. The image dataset used here is a transformation from video to still images in sequence, with a high-speed digital camera used to record or capture multiple images of material/specimen deformation. This will result in the identification of a more precise value. The image data set is then processed to produce a more meaningful image quality. The correlation values can be defined from the post-process image, and after the values are plotted, there is an increase and decrease until the correlation value becomes the highest value. It demonstrates that there is an expansion of the molecules, causing the specimen to break.

Keywords:- Digital Image Correlation , DIC technique ,strain field, post-process image

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Introduction

Digital Image Correlation (DIC) is the process of registering two (or more) images of the same scene and extracting the displacement fields that allow the best match to be obtained. DIC is the solid mechanics equivalent of particle image velocimetry (PIV), which is used in fluid mechanics. DIC was launched a few years after PIV. However, because the sought displacement and strain resolutions were lower than the velocity resolutions, its development was slowed because the algorithmic challenges were more difficult to overcome. Today, it is possible to state that the performance of DIC techniques enables the experimentalist to use them in the majority of practical cases.

In many engineering applications, it is essential to detect displacement or deformation at any location in a structural element or material subject to external stresses. The generated strain values are used to illustrate the structural member's strength problems. Extensometers and strain gauges, which are frequently used instruments for measuring strain, cannot produce strain maps. The Digital Image Correlation (DIC) technique allows for the assessment of full-field strain. This approach is a subset of non-interferometric optical techniques that compare surface image alterations of a test item before and after deformation to determine the degree of deformation. The DIC approach enables innovative and sophisticated studies.

Pre-processing, image enhancement, image modification, and image classification are the four categories into which image processing operations are separated. After one or more processes have been applied to the image, the image quality will be higher than the raw/original image, allowing it to be used for pixel image analysis and other purposes. DIC is an optical technique that uses image tracking and registration methods to accurately measure changes in images in 2D and 3D. This technique is commonly used to evaluate full-field displacement and strains. As a result, it will be able to provide accurate measurements in a study with the help of image processing.

The DIC method allows for full-field strain measurement. This technique is one of a class of non-interferometric optical methods for measuring deformation by comparing 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 calculates displacement and strain using a correlation algorithm.

1.2 Problem statement

Many scientific and engineering disciplines employ the digital image correlation approach to detect displacement. Figure 1 depicts a high-speed digital camera, which will be used to record or capture multiple images of material/specimen deformation. It can, however, be less accurate than traditional testing, especially if the camera is hit or moved during the test.

They also stated that traditional testing is not recommended because it necessitates numerous additional procedures when using a physical strain gauge. This type of testing is known as conventional strain gauges.

Figure 1: UTM Machine

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

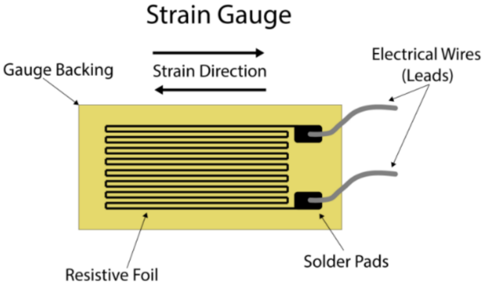

Figure 2: Conventional Strain Gauges

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

This project's literature review focuses on the findings and discussions of several research papers and journals. The primary goal of this literature review and investigation is to guide the project with references to credible and credible sources. This project's primary methodology is digital image correlation (DIC). Accurate characterization of the material under load is made possible by DIC's capacity to capture pictures of distorted surfaces. The strain distribution is produced when the derivatives are applied to the displacement field. On the surface of the specimen, several techniques can be drawn, including dots, grids, and lines, to achieve this. The resolution of the camera and the calibre of the speckle pattern on the material are the two main determinants of the quality of the findings. To get the intended outcome, some picture pre-processing will be applied to the collected photos.

2.2 Summary and Findings

[1] The Novel Digital Image Correlation Technique in Predicting Behaviour and Failure of Hybrid Composite

A technique for detecting deformation on in-plane hybrid CFRP/GFRP composites is described in this study. The use of Digital Image Correlation (DIC) technology facilitates the challenging task of determining the mechanical properties of the hybrid composite. Ultimate Tensile Strength (UTS), Yield Strength (σy), Elastic Modulus (E), Poisson Ratio (v), and Percent Elongation (ΔL) are the major parameters that may be measured during a tensile test.

The framework they employed was Sample Preparation, through which they obtained the modulus of elasticity in the longitudinal direction, E1, for CFRP is roughly 120GPa, and they obtained Formatting Displacements in this. According to ASTM D3039, the modulus of elasticity for GFRP unidirectional 0 degree is in the region of 40 to 45GPa. Furthermore, the user must enter and configure the displacement format for the Ncorr platform, including measurement calibration and scale. These settings were used to transform the displacements from pixels to real units.

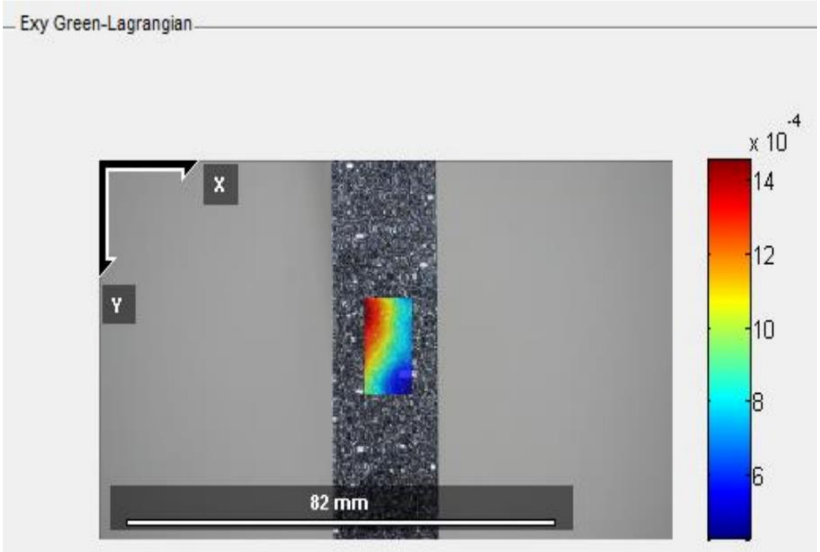

Figure 3: strain countour for eyy which average at 0.0018 for 4000N

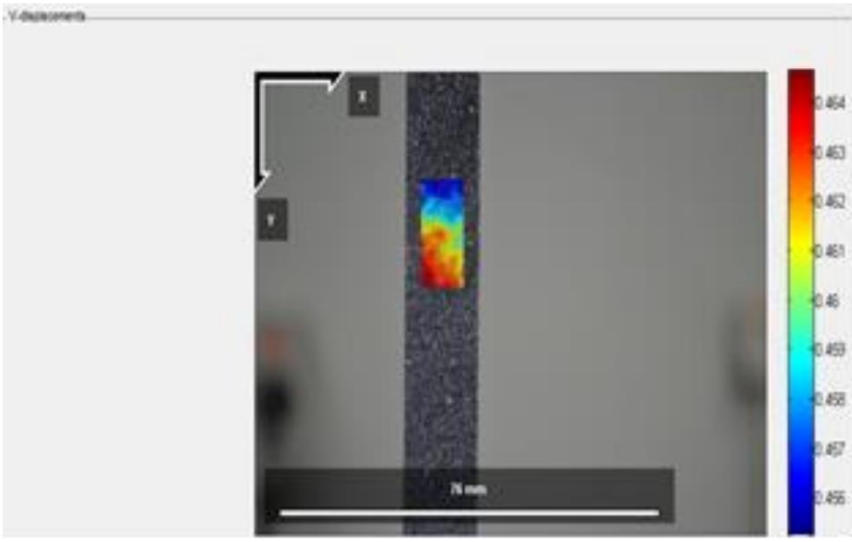

Figure 4: Countour of strain,eyy which average at 0.003 correspond to 7000N

[2] Digital Image Correlation Technique for Strain Measurement of Aluminium Plate

This study describes the Digital Image Correlation Technique for Aluminium Plate Strain Measurement. Digital image correlation (DIC), a new experimental approach for stress-strain research, has been presented. DIC, unlike other optical technologies, does not require any pre-treatment and may be evaluated on virtually any material with a wide range of investigation.

DIC generates strain fields by analysing the movement of designated dots on the subject using the image correlation principle.

The Digital Image Correlation (DIC) approach is cutting-edge for precise strain assessment. Because it provides for fast data collection, this method is appropriate for characterising material properties in the elastic and plastic ranges. It also offers whole field, noncontact, and exceptionally high precision for monitoring displacement and strain. The findings are obtained in the form of graphs from both the tensile test performed on UTM and the DIC technique, which is implemented in MATLAB.

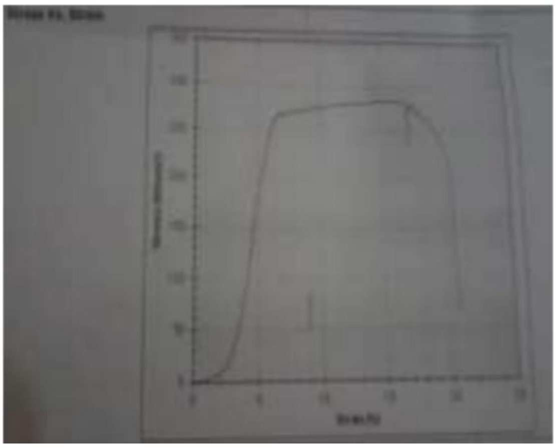

Figure 5: Graph of strain versus stress obtain from UTM

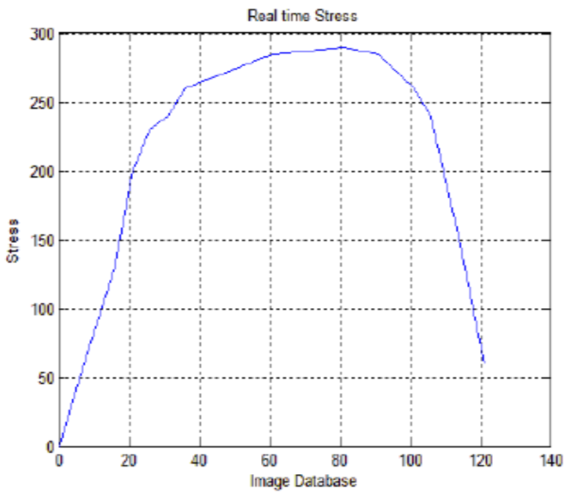

Figure 6: Graph of stress versus image database obtain from DIC software

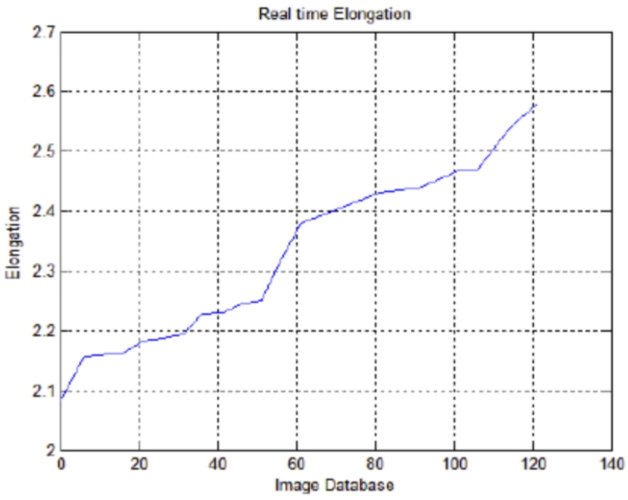

Figure 7: Graph of Elongation versus image database obtain from DIC software

[3] Application of Digital Image Correlation (DIC) Method for Road Material Testing

This study presents the structure of the digital image correlation (DIC) technique, as well as its merits and downsides. The DIC technique is a non-contact, non-interferometric optical approach for detecting the surface deformation of structural components and material samples that may be used in static and fatigue testing. The experimental component of the paper discusses the application of the DIC method for various building material laboratory tests. The results of studies on samples of various materials used in road construction, such as asphalt mixtures (HMA), stone, soil stabilised with a hydraulic binder, and geosynthetics, are discussed. The study's findings demonstrated that, while taking into account their uniqueness, the DIC technique may be used to measure the deformation of road materials in laboratory trials. Because of the variety of test samples, it was possible to identify the places where the DIC method and the algorithms used to analyse the results gave a significant advantage over tensometric measurement methods.

[4] Digital Image Correlation (DIC) Technique in Measuring Strain Using Opensource Platform Ncorr

The open-source platform Ncorr is used in this paper, and the digital image correlation (DIC) technology is described with a logical description of the stages required. The use of digital image correlation in measuring strain is demonstrated beginning with speckle preparation of the gauge area's surface, followed by image acquisition using video imaging, image splitting, and image correlation post-processing to produce the displacement and strain field on the surface under investigation. The quality of the outputs will be evaluated, as will specific technical challenges. Tensile testing on GFRP was utilised to evaluate the strain measurement result obtained using Ncorr, and the experimentally determined modulus of elasticity was compared. The limitations of using in-house equipment such as DSLR cameras with significantly lower frame per second capability and resolution have made the prospect of high reliability low-cost strain measurement using high speed camera and integrated load cell data acquisition for recommendations and future research activity possible.

[5] Speckle patterns for DIC in challenging scenarios: rapid application and impact endurance

They came to the conclusion that commercially available temporary tattoos may be utilised to generate high-quality, tailored, and optimised "tattoo" speckle patterns that can be applied to specimens quickly. They presented practical, inventive, simple-to-implement, and low-cost solutions to difficult circumstances in which speckle patterns could not be created using standard approaches. The adaptability of the "tattoo" dot was successfully shown in two distinct materials (GFRP and aluminium) and stacking systems (semi static and high strain rate), demonstrating excellent bonding and the ability to withstand tremendous stresses. For impact testing on large composite panels, speckle stamping and a thin epoxy adhesive coating were found to perform effectively and save time, whilst patterns formed with more traditional techniques would generally tear off under dynamic stress.

[6] Ultra High Speed DIC and Virtual Fields Method Analysis of a Three Point Bending Impact Test on an Aluminium Bar

In this study, a Hopkinson bar device is employed to strike an aluminium beam in a three-point bending configuration. Digital image correlation was utilised to assess full-field distortion on images captured by an ultra-high-speed camera (16 frames at a temporal resolution of 10µ). The accuracy of the deformation and strain measurements was evaluated, and the data was then quantitatively used to analyse the beam's extraordinarily intricate dynamic behaviour. The interaction between the striker and the flexural bending wave produced by the initial contact was shown to be what governed the beam's deformation. The notion of virtual work was used to reconstruct the impact force from the shear strains and analyse how this impact force related to the specimen's acceleration (inertia forces) and the increase of the bending stresses.

[7] Experimental investigation of the tensile test using digital image correlation (DIC) method

The non-contact optical Digital Image Correlation (DIC) technique is used to quantify contour deformation, strain, and stress. Glass fibre-reinforced plastics and pure resin polymers were all employed to investigate the youthful modulus of aluminium. The DIC method outperforms the other two ways, such as strain gauge measurement and DIC methods. This study used three distinct types of materials to better understand tensile qualities using experimental, strain gauge measurement, and DIC techniques.

[8] Towards High Performance Digital Volume Correlation

This research enhanced the speed and efficiency of the three-dimensional (3D) digital volume correlation (DVC) approach, which analyses displacement and strain fields throughout the interior of a material. The goal is to complete DVC in a fast way while keeping a resolution comparable to that of 2D digital image correlation. This would represent a significant improvement above the state-of-the-art stated in the literature at the time. Creating 3D images with up to 36 billion voxels and resolving features at the 5-micrometre scale using an X-ray micro-CT scanner. We compute twelve degrees of freedom at each correlation point and employ tri cubic spline interpolation to obtain high accuracy. A more effective coarse search, effective spline interpolation, and the use of smoothing splines to manage noisy image data can improve the algorithm's speed and durability. For DVC, data volume, correlation point count, and needed labour to resolve each correlation point all rise cubically. As a result, use parallel computing to deal with the massive increase in computational and memory requirements. They demonstrate the usage of DVC by simulating deformations of 3D micro-CT scans of polymer materials with embedded particles forming an internal pattern.

2.3 Objective

1. To propose a framework for digital image correlation.
2. To obtain an accurate analysis of the material under load.

2.4 Expected output

This project's expected output is the ability to investigate and evaluate the most recent research, successfully propose an image processing-based framework for digital image correlation, and assess the accuracy of the analysis on the material under stress.

CHAPTER 3

PROJECT METHODOLOGY

3.1 Chapter Introduction

This chapter will go over the methodology that will be used for this project. Methodology describes how this project was carried out, the methods used, and the reasons for selecting those methods.

3.2 Architecture of the System

The components are shown in the picture below, each component of the process will be covered in this section.

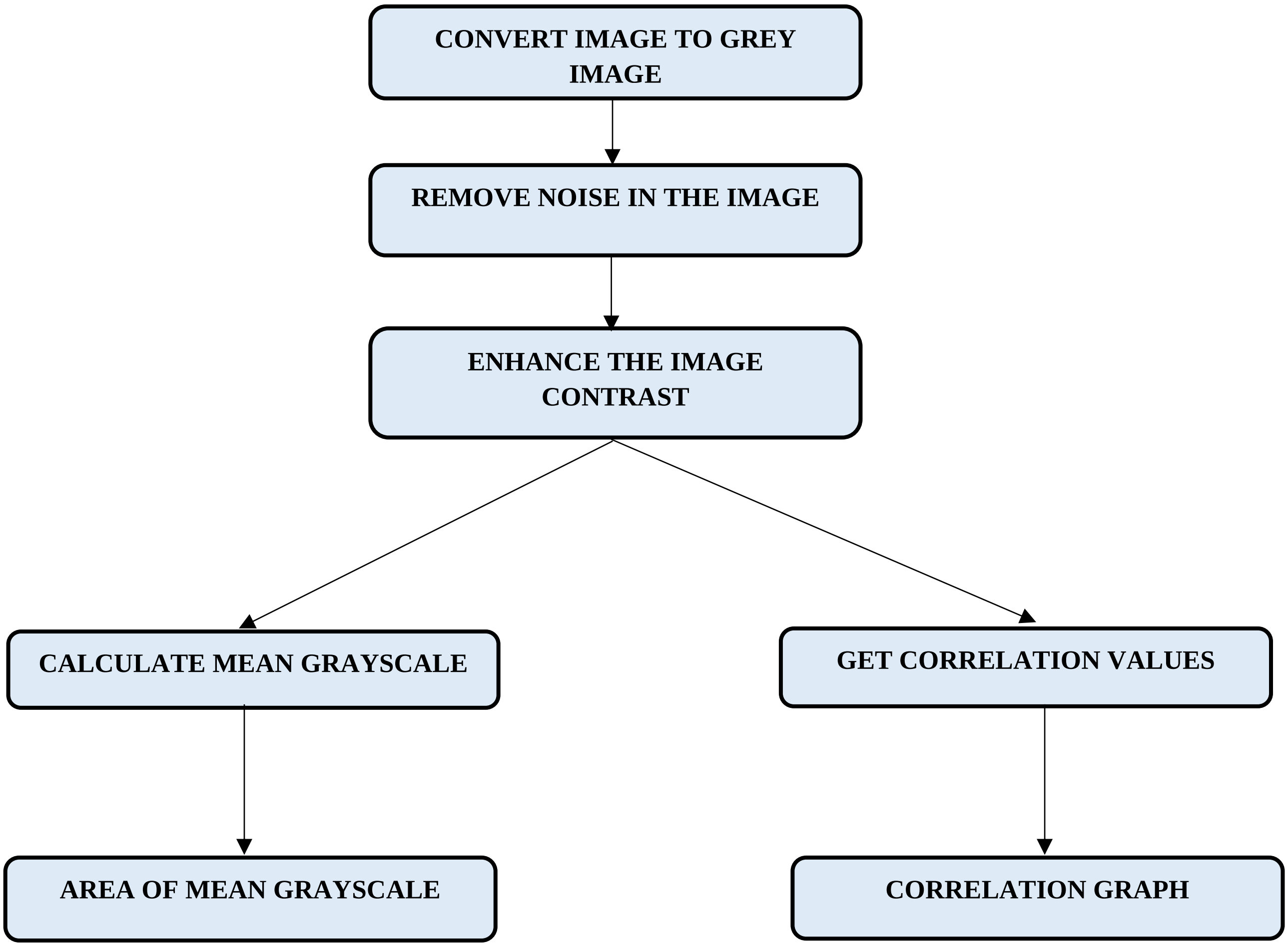

Figure 8: Flowchart of analysis DIC

The first step is to import the database's photos and convert them to grayscale. It is the first picture pre-processing procedure ever carried out. Remove all of the image noise next, and then boost the contrast. These processes in image pre-processing aim to de-noise various forms of noise and enhance the image's visibility. Then we split it into two parts: the first is to determine the grey scale mean's occurrence of elongation and use the area under the grey scale mean graph to verify it; the second is to obtain the correlation value and assess the DIC technique of the image. To make them simpler to analyse, plot the values onto a graph.

Convert image to grey image

The light intensity is represented by each pixel's value in grey scale digital images. In those images the visible range varies from black which is deepest to white which is brightest. To be clear there will be only three colours in image they are grey colour with various shades, White and black in a grayscale image, the value of each pixel is inversely correlated to the number of bits of data utilised to represent it. A greyscale image's value is often represented by 8 bits, or eight binary values combined to represent each pixel's value. As a result, there are 256 different grayscale levels and a value range for pixels of 0-255.

A grayscale image is highly useful for subsequent segmentation processing. All analysis will be performed on the grayscale image in this case. As a result, using equation (1) below, the input RGB image (I) must be transformed to a grayscale image (Ig).

$$Ig = 0.2989 \times IR + 0.5870 \times IG + 0.1140 \times IB \longrightarrow (1)$$

Noise removal

Image noise is an inevitable side effect of image capturing, more simply defined as inaudible yet inescapable fluctuations. Image noise occurs when the light entering the lens misaligns with the sensors of a digital camera. So, in this case, the median filter is applied to the image because it is simple to use and can be used to de-noise various sorts of disturbances. Figure below shows an example of this technique.

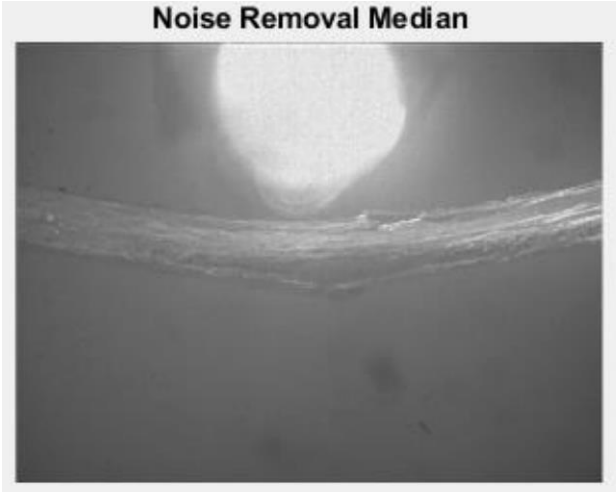

Figure 9: Example of Noise Removal

Contrast enhancement

To improve visibility, contrast enhancement procedures modify the relative brightness and darkness of items in the scene. The image's contrast and tone can be altered by mapping the image's grey levels to new values using a graylevel transform. It is an important aspect in any subjective picture quality evaluation that is used to improve the overall quality of the medical image for feature visualisation and clinical measurement. As a result, it is critical to boost the contrast of such images before further processing and analysis. The figure below shows the image after it has been enhanced for contrast.

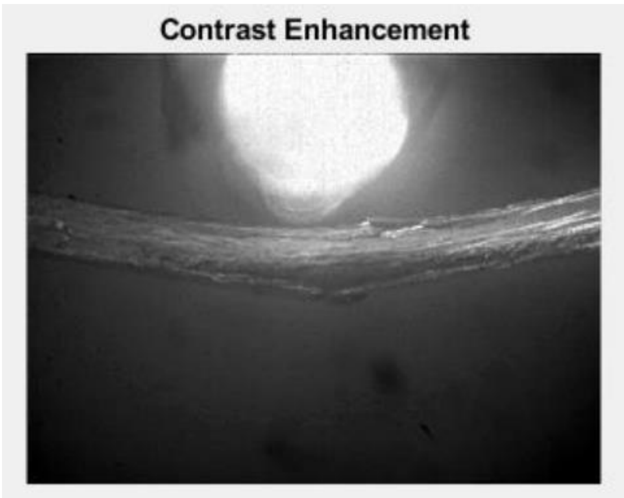

Figure 10: Example of Contrast Enhancement

Calculate Mean Grayscale

Elongation is the amount of distortion that happens before a material break when subjected to a tensile force. After completing the pre-processing stage of this analysis, the Region of Interest (ROI) must be defined, as illustrated in Figure. The most important thing to remember here is that the ROI must be the same size layout for all Images.

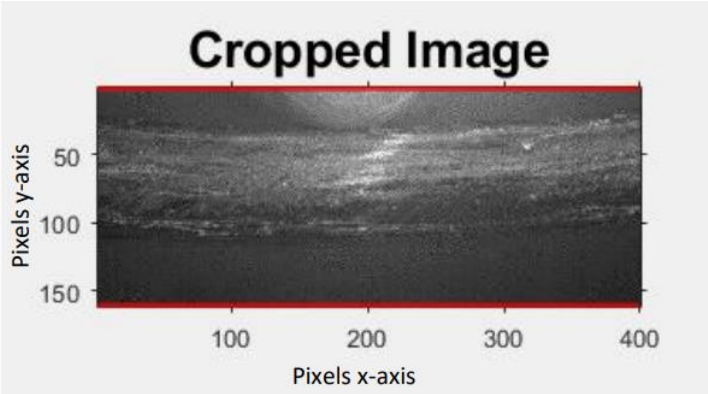

Figure 11: Cropped Image

The image coefficient is then computed. It is depicted in the figure below. Because grayscale has a value range of 0-255 pixels, the mean is calculated by dividing each scale present in a particular image by its sum. The mean was plotted using the blue line colour in Figure 12. The red line represents the greatest value of each point, while the green line is a curve to fit the data based on the coefficients discovered. The coding will be used to explain this further in the process.

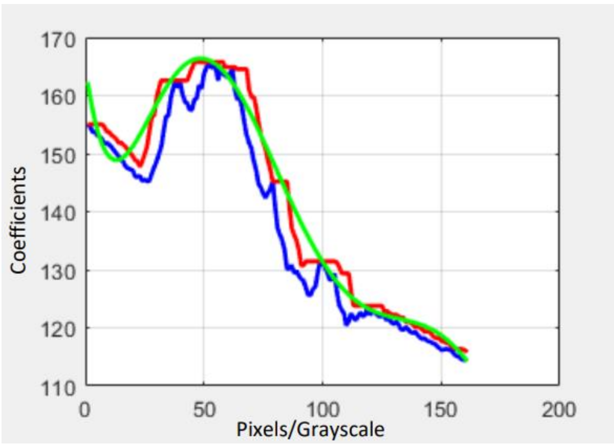

Figure 12: Coefficients

The cropped image is used to define the image's threshold width. As illustrated in Figure below, the background corrected image is used to determine the width.

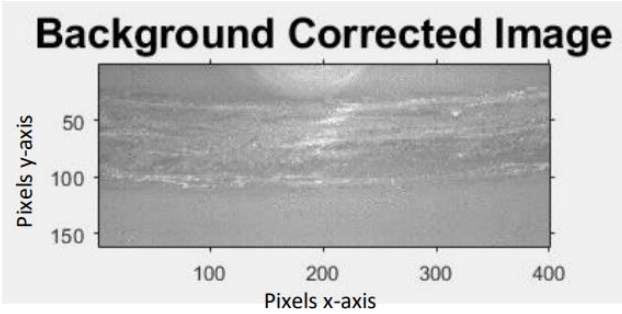

Figure 13: Background Corrected Image

Repeat this method for each image dataset until you receive the results displayed in the figure 14 below, which shows the mean grayscale of a background corrected image. The x-axis represents the image pixels, while the y-axis represents the mean values. Get the values and plot them all in one graph in Excel. Later, the elongation can be seen.

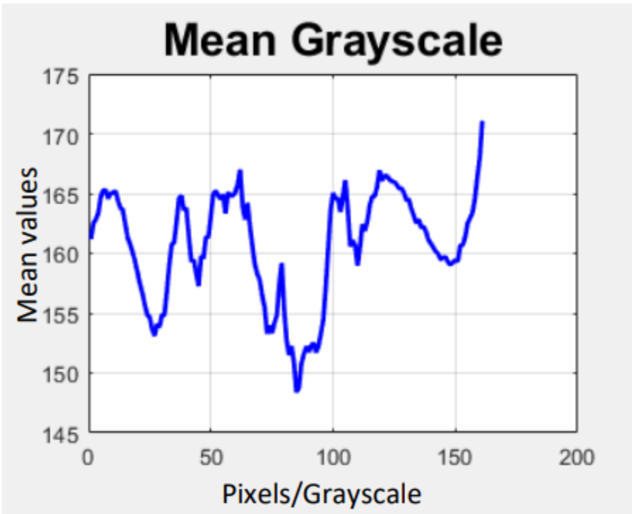

Figure 14: Graph of Width

Correlation Values

A Gray-level co-occurrence matrix feature is called Graycoprops. The syntax that will be utilised to determine the correlation values is listed below.

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

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

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

Figure 15: How to find correlation values

The values that we will see in the command window are shown in Figure 13 above. Only the correlation value is utilised to eventually put all of the photos into a graph.

Area of Mean Grayscale

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

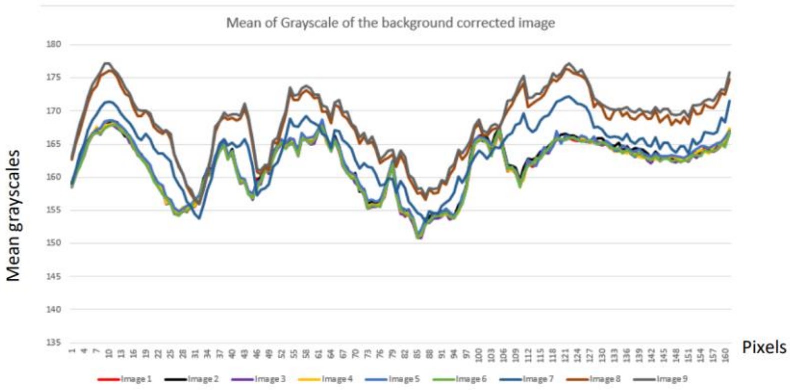

Figure 16: Graph of mean grayscale of the background corrected images

The mean value increases from one image to the next in Figure 16 above. Excel is used to plot all of the image's mean values in a single graph that is challenging to read with the unaided eye. Therefore, it has been demonstrated by the area under the mean grayscale graph in Figure 17.

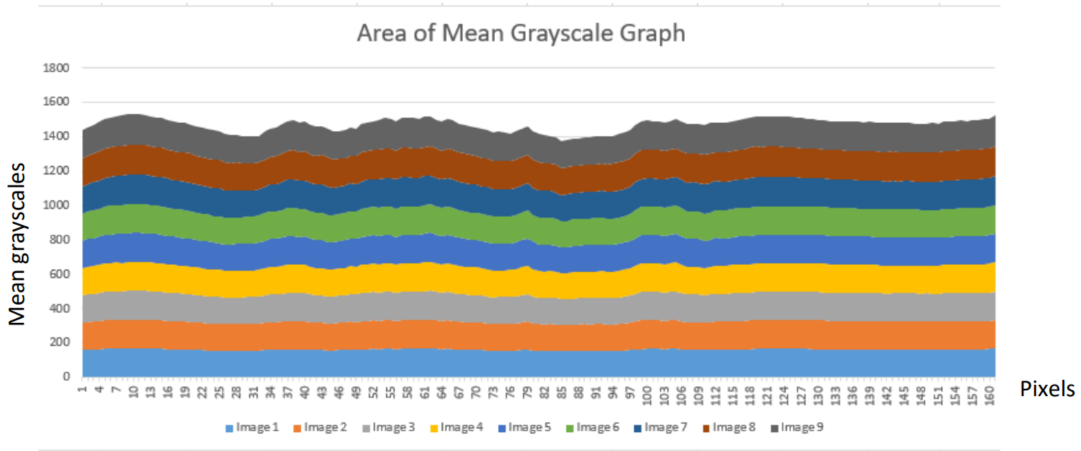

Figure 17: Area of Mean Grayscale Graph

From the above finding, it may be inferred that elongation does really occur.

Correlation Graph

The correlation values that have previously been graphed are displayed below.

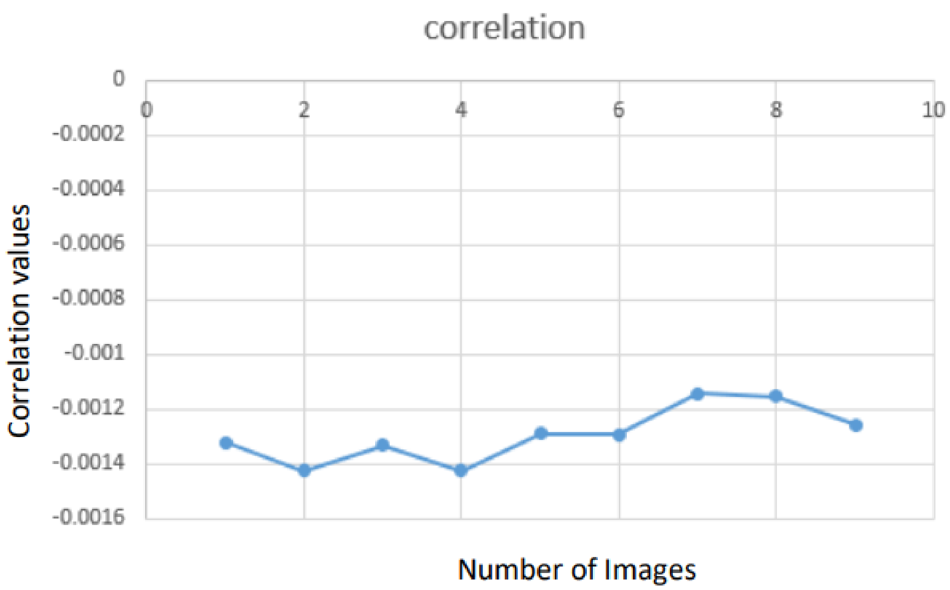

Figure 18: Correlation Graph

The plots of correlation values obtained from all of the photos are shown in Figure 18. The points are plotted at negative values, as may be seen.

3.3 Dataset

Images are taken with the help of a high-speed Digital Camera which has resolution of 16: 9 aspect ratios.

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.

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.

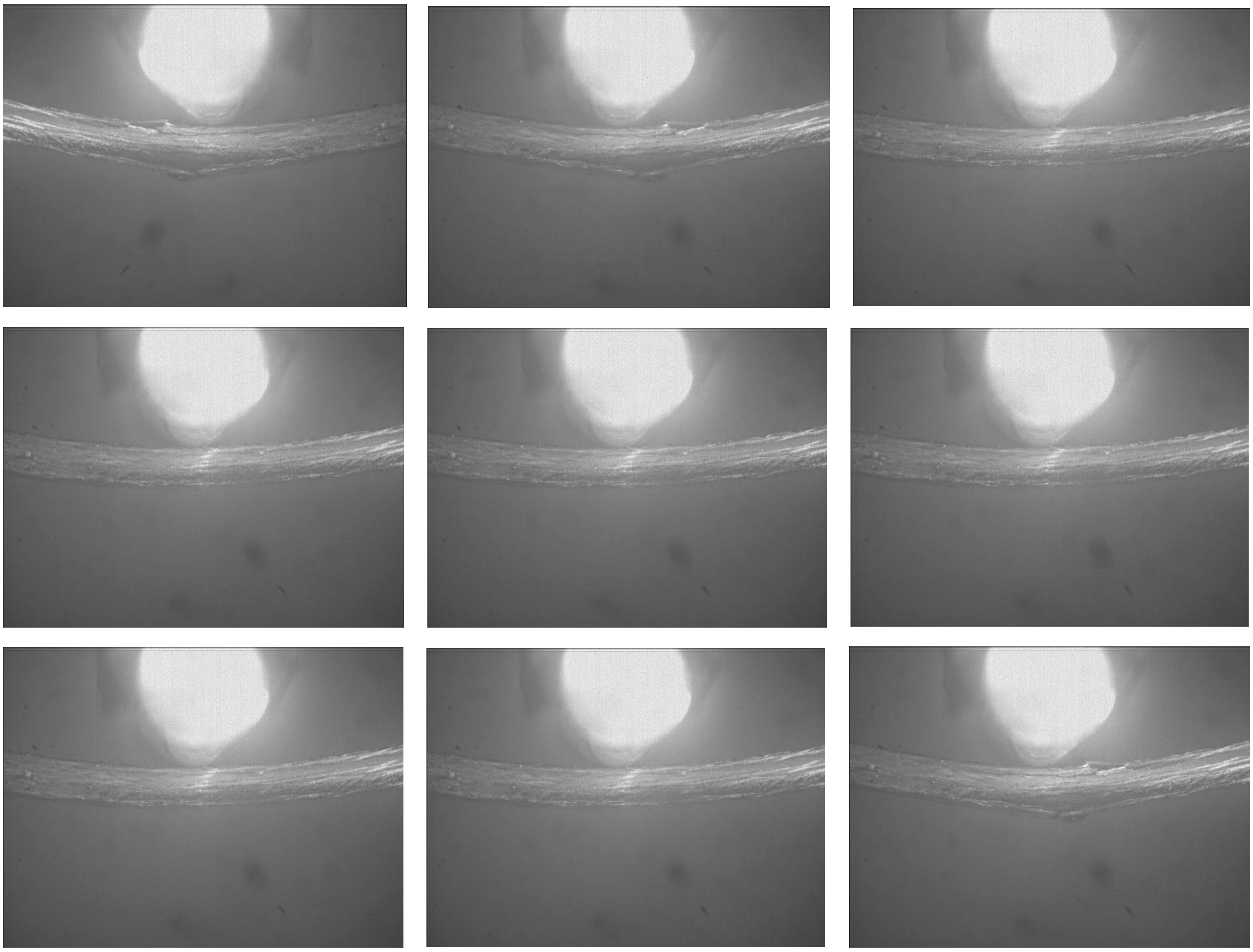

Fig. Images from the database

CHAPTER 4

Network System Architecture

The system's Graphic User Interface (GUI) will essentially allow users to choose the DIC picture from a file (database). All photos will undergo an image preprocessing stage after choosing image samples. A performance evaluation is conducted, and the results are graphed. Analyze the outcomes after that. The data flow diagram that will be used is shown in Figure 17 below.

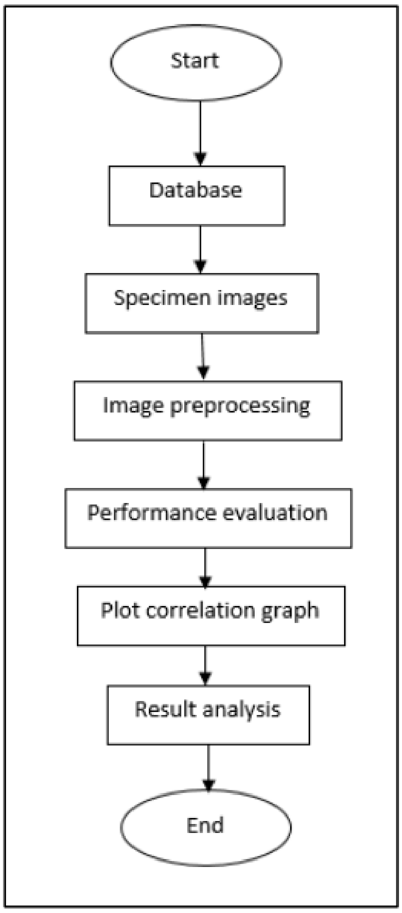

Figure 19: Data Flow Diagram of the system

4.1 Study of Possible Scenarios

The majority of earlier study papers only presented the DIC data in picture form.There aren't many easily accessible research papers on DIC, and none of them have displayed the correlation coefficient data for the DIC images. This is due to DIC being a relatively new industrial phenomenon. By inserting the image into a GUI that will be made, it will be easy in finding the elongation or correlation coefficient of the DIC image.

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

Figure 20: Read and convert to Gray Image

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

Figure 21: Noise Removal

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

Figure 22: Contrast Enhancement

The code for noise reduction, contrast enhancement, and grayscale picture conversion is shown in the figure above, accordingly. In order to interpret the figures above, we must first read the necessary picture from the database and then convert it to a grayscale image using the 'rgb2gray ()' method. Grayscale the image we created with "imshow()" and display it. The same holds true for photographs retrieved from the database, which are designated as Original Images so that it is simple to compare all of the images. Next, remove any extra noise from the Grayscale image using the'medfilt2()' function, and give the result the name Noise Removal Median. Finally, add contrast enhancement to the Noise Removal Image by enhancing the image contrast with the 'imadjust()' method and giving it the name Contrast Enhancement. The Original Picture, Grayscale, Noise Removal Median, and Contrast Enhancement image findings are shown in the figure below.

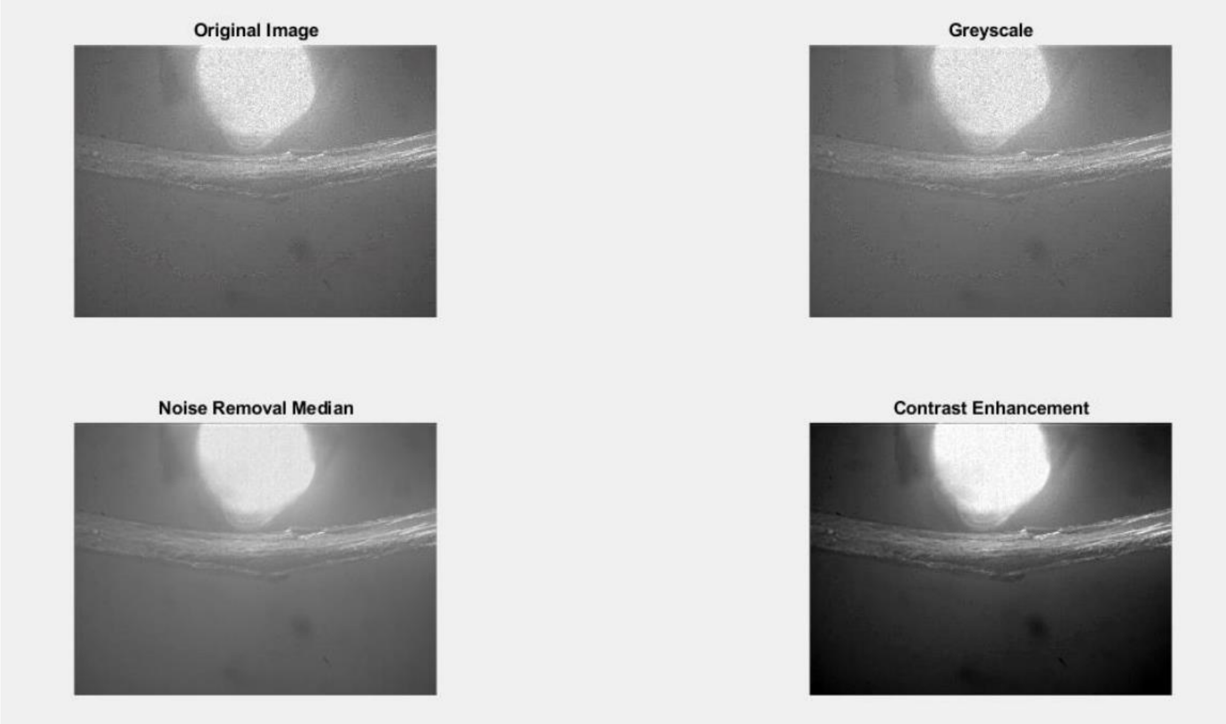

Figure 23: Preprocessing Image

Next, image is cropped to get the ROI. we displaying it through 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)
graylmage = grayImage(rows1:rows2, col1:col2);
```

Figure 24: How to crop the desired segment

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

Figure 25: Display cropped image

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

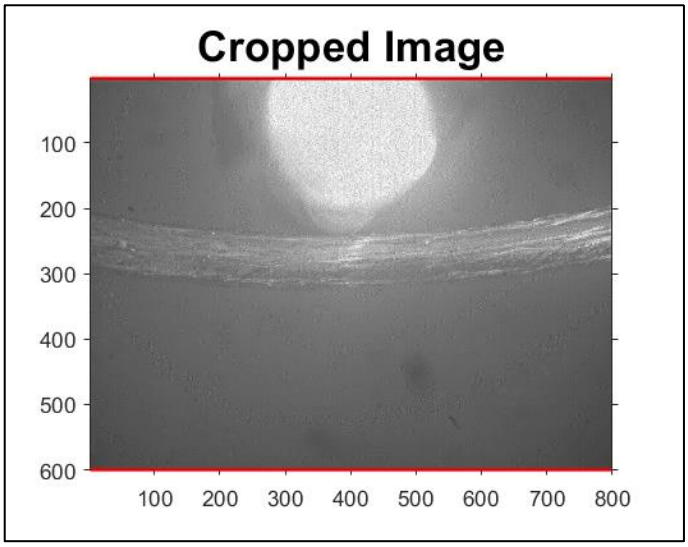

Figure 26: Result of displaying cropped image

4.2 Metric Measurement

Here, we'll go through the metrics calculation in a little more detail. Get the grayscale image's value first, then calculate its mean. derived from Figure The mean was represented by the blue line in the graph. Then, using "movmax," determine the highest values at each position, and plot those values in red. The "grid on" and "hold on" commands display the grid of lines on the graph and hold the plot line and the other plot line together.

```
%% 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);
```

Figure 27: Calculate the coefficients of the image

The polynomial coefficients generated by "Polyfit" in Figure 28 can be used to model a curve and fit the data. A polynomial is evaluated using the "Polyval" for a given set of x values. In order to fit the data, Polyval creates a curve based on the coefficients discovered by polyfit.

```
% 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);
```

Figure 28: Plot the value to the graph

The outcome of the mean vs. grayscale graph, which represents the coefficient of the picture, is shown in Figure 29 below. One graph has the mean (blue), maximum (red), and polynomial (green) plot lines.

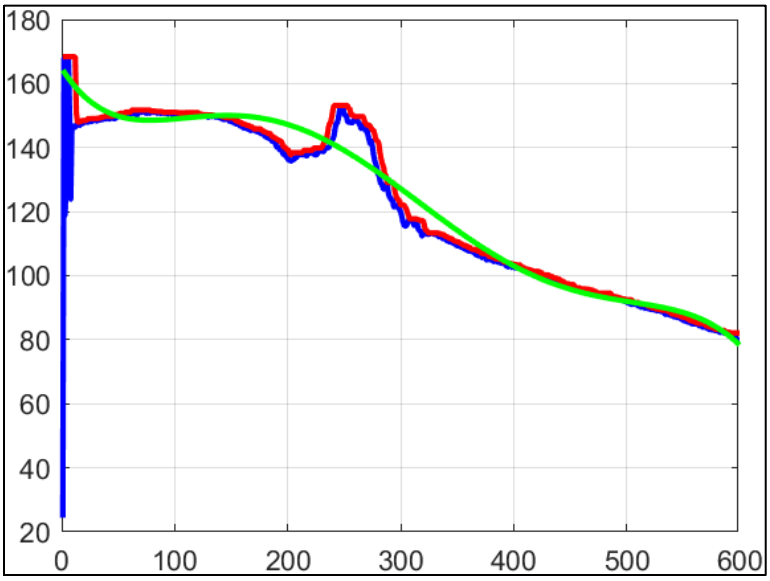

Figure 29: Graph of coefficients of the image

Figure 30 shows how to make the backdrop picture. Background detection is a popular method for detecting moving objects in a succession of pictures from static cameras. This method is based on recognising moving objects based on the difference between the current frame and the reference frame.

So, in order to produce the background picture, we must display the background Image with pixels containing grayscale values ranging from 0 to 255. Call it 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 30: Create a background image

Locate the background-corrected image next. The technique of perfectly matching a picture's projection to a given projection surface or form involves digitally modifying image data. Figure 30's background picture will be layered with the grey image to create Figure 32's background-corrected image. Figure 31's Equation (2) demonstrates how to overlay the background picture with the grey image. the result of the Background Corrected Image procedure.

$$grayImage = unit8\left(\max\left(backgroundImage(:)\right) * double(grayImage)\ .\!/\ backgroundImage\right) \quad \textbf{(2)}$$

```
% 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 31: Display the background corrected image

Figure 32 shows the outcome of this procedure. We can see that it displays the pixel information on the image's x- and y-axes.

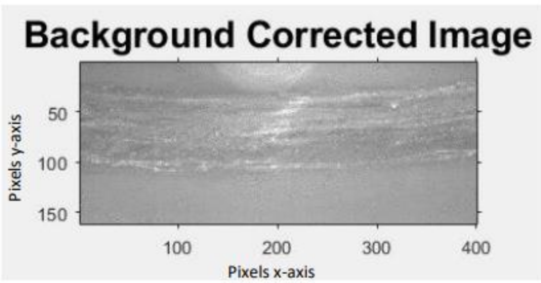

Figure 32: Results of background corrected image

Find the most recent mean grey of the background-corrected picture in Figure 33 using the "mean()" function, then graph it with the blue line.

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

Figure 33: The Mean Grayscale codes

The restored image's mean grayscale is displayed in Figure 34. We can see that it displays mean grayscale values at the y-axis and pixel values at the x-axis.

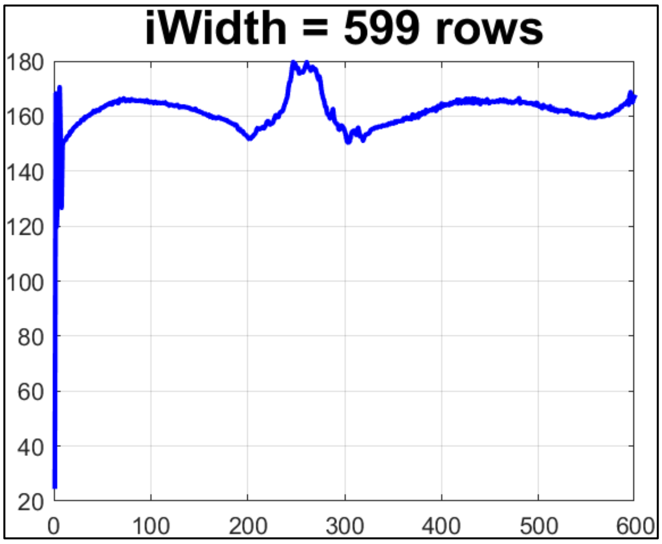

Figure 34: Graph of width

Except for the GUI code, all of the analytical coding is given in this chapter. It will assist a lot of scholars locate fresh information on DIC because it is still new in this subject and there aren't a lot of study papers on it.

CHAPTER 5

IMPLEMENTATION

The implementation between the processing codes and the Matlab-created GUI will be described in this chapter.

5.1 Environment Setup

Drag-and-drop design and an integrated editor allow for the creation of the graphical user interface (GUI), which can rapidly code its functionality.

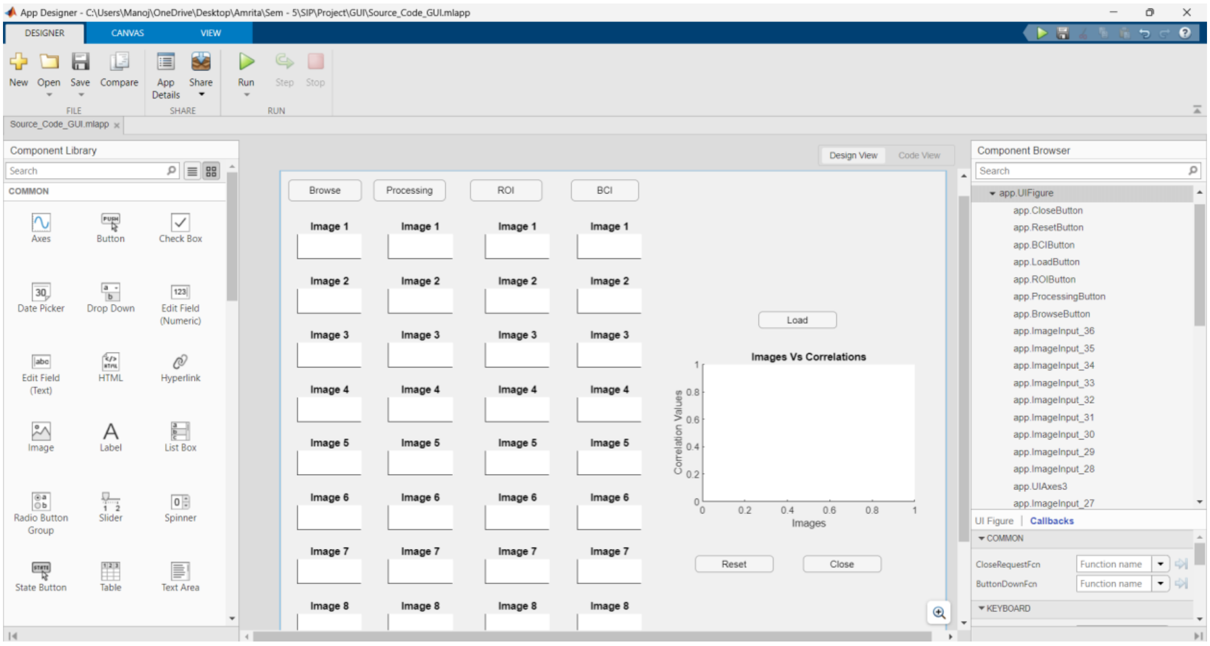

Figure 35: App Designer

Figure 35 shows GUI after all necessary components were dropped into place. The button and the axis are the parts that are utilized the most. Once the designing was complete, it was necessary to connect the button and axis components. Therefore, after the appropriate image has been summoned, it may be processed by simply clicking the button that has been provided.

```
% Button pushed function: BrowseButton
function BrowseButtonPushed(app, event)

prompt = {'Enater Image No.:'};
dlgtitle = 'Input Image';
dims = [1 35];
answer = inputdlg(prompt,dlgtitle,dims);
ImgId = answer{1};
name_ = "Pick Image "+ImgId

if app.ImageInput_1.Title.String == "Image "+ImgId

        [filename, pathname] = uigetfile('*.*',name_);
         if isequal(filename,0) || isequal(pathname,0)
                 helpdlg('User pressed cancel');
         else
              InputImage=imread([pathname, filename]);
              imshow(InputImage, 'Parent',app.ImageInput_1);
              app.InputImage1=InputImage;
         end
```

Figure 36: Code of Browse Button pushed function

Right-click the button you want to start first and callback in the component browser. The Code View will be brought to the state depicted in Figure 36.

To call the image from the existing file, line 88–93 utilises the same Matlab code.As seen in Figure 36 at line 93, connect the button to the required axis. The same code should be repeated until all the desired photos can be called.

Follow the same procedures as previously for each button on the GUI. The codes for the Processing, ROI, and BCI buttons are shown in Figures 37, 38, and 39, respectively. The codes used here are all identical Matlab codes. Then, as illustrated at lines 191, 296, and 352, link each button to the appropriate axis. The same code should be repeated until all the desired photos can be called.

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

app.InputImage1 = app.InputImage1;
[~, ~, numberOfColorChannels] = size(app.InputImage1);
if numberOfColorChannels > 1
    % It's not really gray s
    % cale like we expected - it's color.
    % Extract the red channel (so the magenta lines will be white).
    app.InputImage1 = app.InputImage1(:, :, 1);
end
Kmedian1 = medfilt2(app.InputImage1); %preprocessing
app.Contrast1 = imadjust(Kmedian1);
imshow(app.Contrast1,'Parent',app.ImageInput_10)
app.Contrast1=app.Contrast1;
```

Figure 37: Code of Processing Button pushed function

```
% Button pushed function: ROIButton
function ROIButtonPushed(app, event)
    app.Contrast1 = app.Contrast1;
app.CropIm1 = imcrop(app.Contrast1,[200 210 400 160]); %[xmin ymin xlenght ylenght] | x=col y=row
imshow(app.CropIm1,'Parent',app.ImageInput_19)
app.CropIm1=app.CropIm1;
```

Figure 38: Code of ROI Button pushed function

```
% Button pushed function: BCIButton
function BCIButtonPushed(app, event)
    app.CropIm1=app.CropIm1;
    verticalProfile = mean(app.CropIm1, 2); % Mean of Matrix Row
    verticalProfile = movmax(verticalProfile, 13); %moving maximum
    xFit = 1 : length(verticalProfile);
    coefficients = polyfit(xFit, verticalProfile, 6); %coefficients
    [~, columns] = size(app.CropIm1);
    yFit = polyval(coefficients, xFit); %polyval
    backgroundImage = repmat(yFit(:), [1, columns]);
    B1 = uint8(max(backgroundImage(:)) * double(app.CropIm1) ./ backgroundImage);
    imshow(B1,'Parent',app.ImageInput_28);
```

Figure 39: Code of BCI Button pushed funtion

CHAPTER 6

RESULTS AND INFERENCES

In this chapter, we'll display the graphical user interface (GUI) graph that represents the work we did throughout the implementation stage.

6.1 Results and Analysis

Figure 40 below displays the outcome following the selection of the ROI and Background Corrected Image during image processing (BCI).

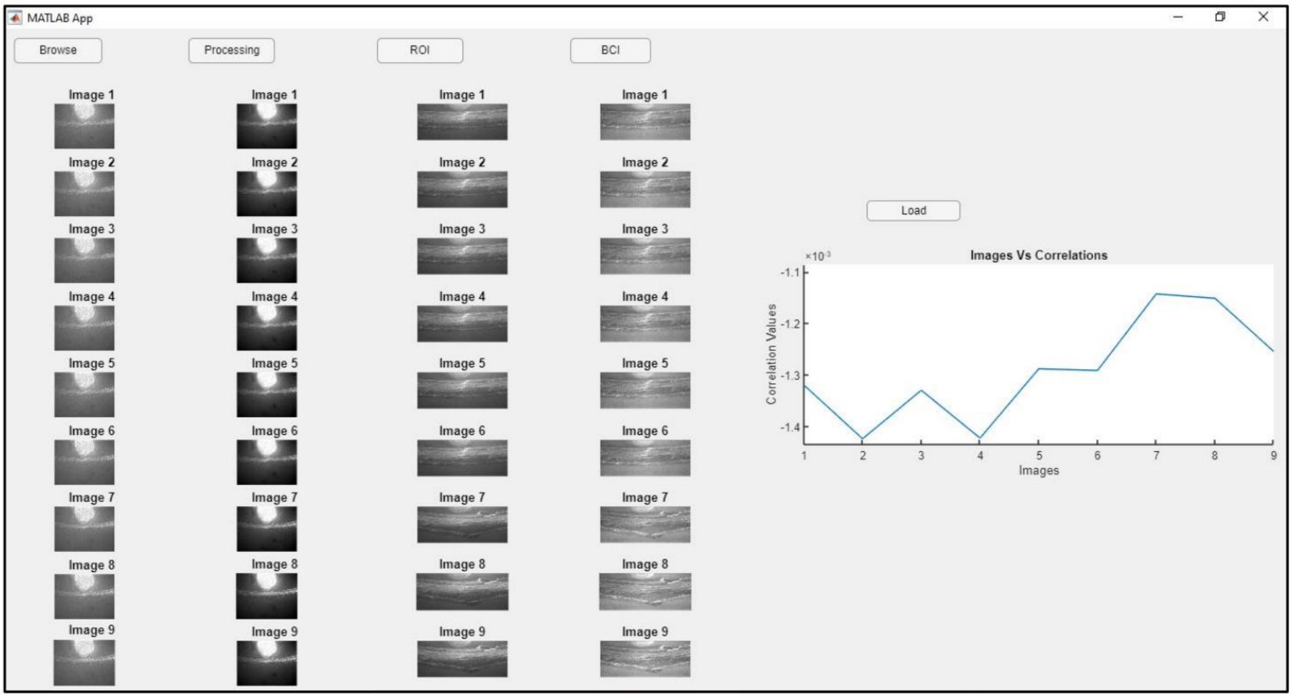

Figure 40: GUI of DIC technique using Image Preprocessing

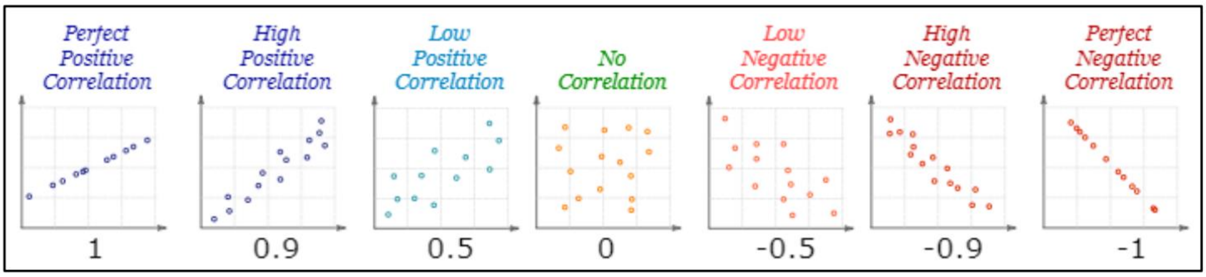

Figure 41: Correlation Graph

The values of the correlation will be obtained using the BCI pictures. Following that, the values will be presented on a graph of Image Vs. Correlation.

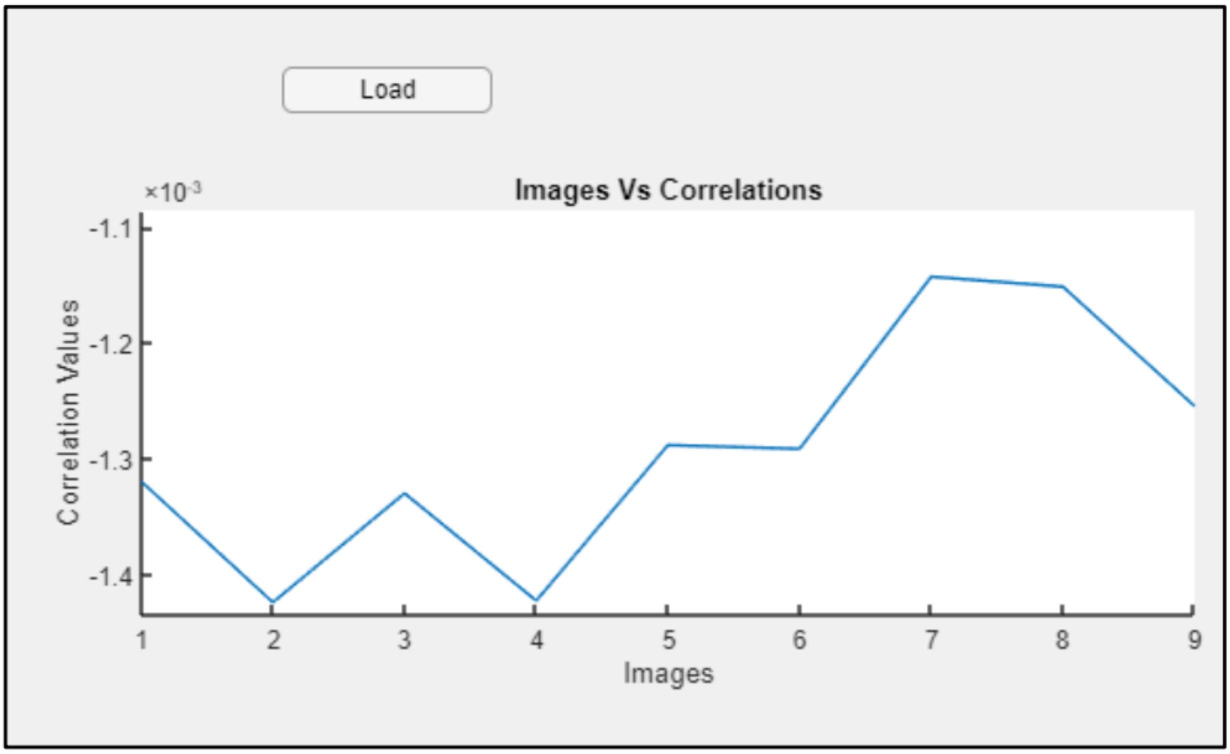

Figure 42: Images Vs Correlations

we abstracted the numbers from the excel file and displayed them on the graph. This is one of the system's limitations it still can't determine the correlation value straight from the image and proceed to put it on a graph. Therefore, we manually locate it in the Matlab command window using the "Graycoprops()" function, copy the correlation values, and paste them into the "correlation" cell of Excel.

The correlations calculated from the photographs have a negative value, as illustrated in Figure 42. As successive photographs are analysed, the values rise and fall until the correlation value reaches its greatest value in the seventh image. It is the number that is most similar to 0.

In Figure 41, the molecules spread as the negative correlation value approaches correlation 0. It may be considered to be expanding, in other terms. Since of this, the specimen might shatter because there is not a strong enough association between the molecules.

Figure 43: Zoom in to Image 7 where it starts to break

In this chapter, we examined the relationship between the sought correlation value and the specimen's state on the provided picture collection.

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENTS

This chapter is where we will conclude up the entire project. There are some more restrictions, which will be discussed below.

7.1 Summary of the work

The system was created and constructed using MATLAB's AppDesigner to use image processing to produce correlation values. We can evaluate why the specimen in the picture collection can break using those values.

7.2 Contributions in the work

Traditional Methods of DIC – Neelam Venkata Sai Manoj

Literature Survey - P V R Subba Rao

System Architecture - Krishna Jayanth K

Analysis Of Correlations - Venkata Ramana Murthy P

Graphical User Interface - Mohith V

7.3 Future Works

Several limits were found throughout the study, as was already discussed. The first is that once photographs are submitted, they cannot be cancelled or deleted. If the image order is entered incorrectly, it is also not able to remove it. Next, there are a set number of picture inputs 9 available. They won't be approved and will move on to the next step if there are more than the allotted number. A small number of photographs may also keep the file image location from disappearing and prevent you from moving on to the next step. I think that this system may be improved in the future while dealing with all the restrictions listed, either by adding more graphs or by displaying the required value derived from the image dataset.

CHAPTER 8

References

[1]Ab Ghani, Ahmad Fuad & Mansor, Muhd Ridzuan & Selamat, Mohd & Jumahat, Aidah & Mahmud, Jamaluddin. (2018). The Novel Digital Image Correlation Technique in Predicting Behaviour and Failure of Hybrid Composite.

[2]Gadhe, Supriya & Navthar, Ravindra. (2016). Digital Image Correlation Technique for Strain Measurement of Aluminium Plate. International Journal of Engineering Trends and Technology. 39. 306-311. 10.14445/22315381/IJETT-V39P251.

[3]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

[4]Ab Ghani, Ahmad Fuad & Ali, M. B. & Dhar Malingam, Sivakumar & Mahmud, Jamaluddin. (2016). Digital Image Correlation (DIC) Technique in Measuring Strain Using Opensource Platform Ncorr. Journal of Advanced Research in Applied Mechanics. 26. 2289-7895.

[5] Quino, Gustavo & Chen, Yanhong & Ramakrishnan, Karthik Ram & Martinez-Hergueta, Francisca & Zumpano, Giuseppe & Pellegrino, Antonio & Petrinic, Nik. (2020). Speckle patterns for DIC in challenging scenarios: rapid application and impact endurance. Measurement Science and Technology. 32. 10.1088/1361-6501/abaae8.

[6]Pierron, Fabrice & Sutton, Michael & Tiwari, V.. (2011). Ultra High Speed DIC and Virtual Fields Method Analysis of a Three Point Bending Impact Test on an Aluminium Bar. Experimental Mechanics. 51. 537-563. 10.1007/s11340-010-9402-y.

[7]Ma, Quanjin & Rejab, Ruzaimi & Halim, Qayyum & Merzuki, M.N.M. & Darus, M.A.H.. (2020). Experimental investigation of the tensile test using digital image correlation (DIC) method. Materials Today: Proceedings. 27. 10.1016/j.matpr.2019.12.072.

[8]Gates, M., Lambros, J., & Heath, M. T. (2011). Towards High Performance Digital Volume Correlation. *Experimental Mechanics*, *51*(4), 491-507. https://doi.org/10.1007/s11340-010-9445-0

CHAPTER 9

Appendix

Code for Preprocessing

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

% 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);
```

Code for find elongation

```
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]);
[~, ~, numberOfColorChannels] = size(grayImage);
if numberOfColorChannels > 1
    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.
B = 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('iWidth = %d rows', width);
title(caption, 'FontSize', fontSize, 'Interpreter', 'None');
```