FEATURES EXTRACTION OF HEP-2 IMMUNOFLUORESCENCE PATTERNS BASED ON TEXTURE AND REGION OF INTEREST TECHNIQUES

By

SITI MASTURA BINTI MD HASIM

FINAL PROJECT REPORT

Submitted to Electrical and Electronics Engineering Department
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical and Electronics Engineering)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

FEATURES EXTRACTION OF HEP-2 IMMUNOFLUORESCENCE PATTERNS BASED ON TEXTURE AND REGION-OF-INTEREST TECHNIQUE

By

SITI MASTURA BINTI MD HASIM

13382

A project dissertation submitted to
Electrical and Electronics Engineering Department
Universiti Teknologi PETRONAS
In partial fulfillment of the requirement for
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL AND ELECTRONICS ENGINEERING)

Approved:

_____

Ms Zazilah binti May
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

SEPTEMBER 2013

CERTIFICATE OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project and this project consists of my original work of my own except as specified in the references and acknowledgements and the original work contained in here have not been undertaken or done by unspecified sources of persons.

_____

Siti Mastura binti Md Hasim

ABSTRACT

Autoimmune disease is a disease that happens when improper immune response in the body fighting against substance, cells and tissues that naturally exists and needed in human's body. This will later on cause autoantibody disease such as SLE where internal organ failed to perform their basic functions. Antinuclear antibody (ANA) test is a way to test the presence of autoantibodies in individual blood serum. This study focuses on ANA test that is done using indirect immunofluorescence HEp-2 cell coating slides that are used to place the blood serum. However, there are several problems encountered with current technique, such as inaccuracy of the result as the result is viewed by naked eyes of operator. There is no objective definition for positive, negative or border line of infection. This project involves developing features extraction technique of HEp-2 cell of 2 main patterns namely Nucleolar and Centromere using texture and region of interest technique. Next, to design an algorithm that can automatically identify the 2 patterns of the HEp-2 cell tested using ANA. To execute features extraction, image pre-processing is performed to enhance image in terms of its intensity, brightness and contrast. Only clear and good input image will produce good results. Image segmentation will be done after pre-processing completed to further enhance the image according to its edge or region to be used for the input image. Different methods of features extraction will be used and compared for better outcome. To differentiate between one pattern from another, image classification is done by evaluating the properties of internal image from features extraction and a boundary is drawn between Centromere and Nucleolar pattern. The result shows four different types of properties of internal cells which are homogeneity, contrast, energy and correlation. After analysis has been done, energy between Centromere and Nucleolar are different from each other and used to classify the pattern in SVM classifier. Tools used in this study are MATLAB software and image processing tools in MATLAB.

ACKNOWLEDGEMENT

The author believes that this project would not be able to finish without the contributions and involvement of certain parties, organizations and individuals. In this opportunity, the author would like to express her deepest gratitude to all. First and foremost, thank you to her final year project supervisor, Ms. Zazilah binti May for the continuous guidance, assistance and supervision throughout the progress of Final Year Project. She has constantly provided knowledge and advice for every step since the planning phase of the project until the completion of this dissertation.

The author would like to thank her co-supervisor, Dr. Josefina Barnachea Janier for endless guidance on both technical and non-technical knowledge. Special thanks as well to Dr Nasreen binti Baharuddin and Ms Azrina binti Abd Aziz, FYP coordinators, for conducting several briefings and seminars to assist the FYP students. They had also arranged presentation schedules and invited internal and external examiners to evaluate the students.

Special gratitude to technicians at Hospital Universiti Sains Malaysia (HUSM) for supplying the ANA patterns images to the author in order to do testing and analysis. All in all, the author would love to express her gratitude to supportive colleagues and family for their never-ending support and encouragement throughout the project completion.

Finally, again, the author would like to thank all parties involved directly and indirectly for making this Final Year Project a success.

Table of Content

Table of Figures

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Autoimmune system is improper immune response in which autoantibodies will be developed. Autoantibodies are antibody developed by immune system to fight against body's own protein and will lead to diseases such as systemic lupus erythematosus which is also called as SLE. SLE is an autoimmune disease that will attack patients' cells and tissues at any part of their body. Once affected to this disease, patients will experience multisystem malfunction that can be observed through several symptoms.

In Malaysia, there are 10000 people are getting infected by SLE in every 30 years where 90% of them are women while another 10% are men and children. Out of the 90% of women that are infected, 90% of their age ranges are between 15 to 40 years old [1]. These numbers are based on patients that had been diagnosed.

Antinuclear antibody (ANA) test is a way to test for autoantibodies present in individual blood serum. There are several common tests to detect ANA in blood which either indirect immunofluorescence (IIF) or enzyme-linked immunosorbent assay.

In using indirect immunofluorescence technique, HEp-2 cell is used as a substrate to coat slides, antibody conjugates and reference area [2]. For positive results of ANA, dilution will be done with certain ratio to obtain exact concentration of ANA in that blood sample [3]. In order to detect ANA, the following methods were done.

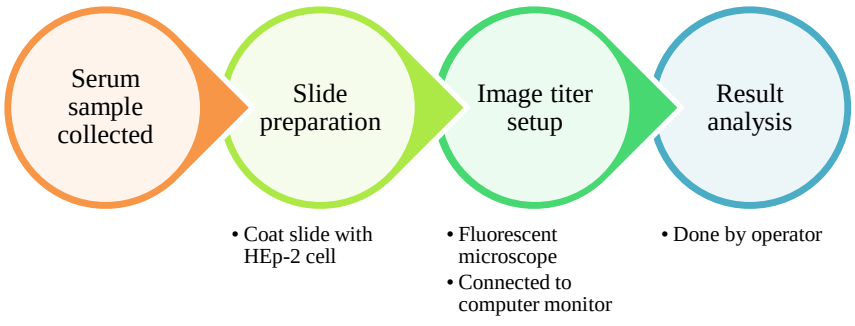

Figure 1: Method to Detect ANA

Serum sample is collected from patients from whom blood is drawn and put on slide coated with HEp-2 cell. Slide was then put under a charge coupled device (CCD) camera in using three separate CCDs in which each one will capture image of separate measurement of primary colors, red, blue and green light [4]. Immunofluorescence image will be obtained and save in the computer connected to the camera. This image results will then be analyzed by 2 operators who will confirm the condition of the patient to be either positive or negative to autoimmune disease.

1.2 Problem Statement

Current technique applied in Malaysia is image analysis done by operator with their naked eyes [2]. Positive, negative or border line of ANA presence in the blood sample can be inaccurately determined.

There are some difficulties in order to distinguish numerous patterns of indirect immunofluorescence HEp-2 cell. Misinterpretation of patterns will result in inaccurate result of diseases. This will affect in wrong treatment to the patient and can cause loss to both patient and the hospital. Significant features of patterns need to be extracted accurately in order to give out precise result to patient.

1.3 Significance of Project

This project aims to develop an algorithm to automatically obtain features of ANA cells that will be used to classify one pattern from another. This will solve the problems of wrongly analyze the types of patterns of ANA by using naked eyes of the two operators and based on their experience.

By having this algorithm, the operators can easily differentiate Centromere pattern from Nucleolar with minimal mistakes which will cause different disease being diagnosed to patients that will lead to wrong treatment been given to them. Therefore, by using this algorithm, patients' lives can be saved and operators can reduce their time taken in analyzing ANA patterns.

1.4 Objective

i. To extract significant features of two patterns (named as Nucleolar and centromere) using texture and region-of-interest technique.

ii. To design an algorithm that will identify these two patterns based on texture and region-of-interest technique.

1.5 Scope of Study

This project is done focusing on two patterns of immunofluorescence HEp-2 cell named as Nucleolar and Centromere. Features extraction will be done in MATLAB software and Image Processing Toolbox concentrating on texture and region-of-interest techniques. For texture technique, GLCM method will be used while for region-of-interest technique, only regularization method will be focused on this project due to time constraint.

1.6 Relevancy of the Project

This project requires the author to do image processing which is included in Data Signal Processing, core syllabus in 4 years of undergraduate study in Electrical and Electronics Engineering. Besides, technical knowledge in MATLAB programming, time management skills and interpersonal communication skills are also required to complete the project such as planning and presentation for the evaluators. As this project focuses on ANA patterns, the author need to do extra researches on medical fields on the diseases which can be applied in real working environment when the company are handling equipment different from electrical and electronics equipment which require the author to become a professional, competent and creative in completing the tasks.

1.7 Feasibility of the Project

The timeline for this project is made for two semesters which is approximately six months to complete. Researches and studies done in the two weeks to grab the main idea of the project and continuously until the project is completed allows the author to be able to understand the project thoroughly. As images are obtained from HUSM, the author able to develop algorithm that is compatible to be used in HUSM once completed. Therefore, the problems occurring to HUSM's medical technicians can be solved and faster treatment can be done towards the patients.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Image casting or capturing will be done using CCD camera where a rigid location of image is selected. Blood sample taken will be put on an illuminated background with HEp-2 cell coating the slide [4]. Once captured, the image will then ready for pre-processing activity. In pre-processing, the image is processed to reduce noise and enhance the image captured to become clearer and providing better input for further image processing later on. Histogram of image will be corrected by either making it flat for better view. Other process such as smoothing and segmentation is done to enhance the image [4, 5].

Image segmentation is a sub process in image processing to divide an image into partition that will reduce gray value variance and threshold. The image is first normalized using equation (1) [6]:

$$G(i,j) = \begin{cases} M_o + \sqrt{\dfrac{V_o(I(i,j) - M_i)^2}{V_i}}, & I(i,j) > M; \\ M_o - \sqrt{\dfrac{V_o(I(i,j) - M_i)^2}{V_i}}, & I(i,j) < M; \end{cases} \quad (1)$$

$$where\ M_o = Mean\ Value\ and\ V_o = Variance\ value$$

Threshold classification is one of the most effective approaches to perform image segmentation [5]. Optimal threshold is obtained by using graythresh instruction with binary value. The result of the image will display portion with less than threshold as contour and gray value greater than threshold will be viewed as white area [4]. Threshold is said to be most operational method due to its small storage space, faster speed and easier in handling.

Another technique is watershed technique where a location marker is set to obtain gradient of image. This technique is good to separate combined object and it has better exactness compared to threshold. However, this technique does not offer smoothing and generalization properties [5].

2.2 Features Extraction

Features extraction is a function to detect and isolate certain fraction or form in input image. Area, perimeter and roundness are the characterization of features [5]. To get better result of features extraction, noise need to be removed in the very first place. This is where the importance of pre-processing takes position. Smoothing and deburring process need to be done with grayscale image before extracting the features [6].

2.2.1 Texture Technique

Texture is defined as natural property of a surface with different visual pattern of images and repeated pixel information in each image. By executing feature extraction, important surface can be separated from its peripheral background [7]. In image processing, texture is also called as pixels or threads. Texture indicates homogeneity patterns in which it results in several color and intensity [8]. There are several methods that can be used to perform features extraction. To calculate features, several formulas are used [7].

$$Energy = \sum_{i,j} P(i,j)^2 \quad (2) \qquad Entropy = -\sum_{i,j} P(i,j) \log P\,(i,j) \quad (3)$$

$$Correlation = \sum_{i,j} \frac{(i-\mu i)(j-\mu j)p\,(i,j)}{\sigma_i \sigma_j} \quad (4) \qquad Homogeneity = \sum_{i,j} \frac{P(i,j)}{1+|i-j|} \quad (5)$$

$$where\ P = number\ of\ occurences\ of\ gray\ leverl\ and\ i,j\ total\ number\ of\ gray\ level\ pairs$$

From the work done in [9], in rotation-invariant method, input image is rotated to certain degree before it was decompose into few sub images by using low pass and band pass filter. Average energy of each sub images is calculated and compared to the largest energy and if $E_k < KxE_{max}$ the decomposition stopped for the particular sub image. However, other sub images need energy calculation to be done. First sub image of low pass filter with highest energy value will be used to calculate its mean and standard deviation.

Based on work in [10], several method was used such as Steerable Pyramid which is similar to orientation-variant method, Contourlet Transform, Gabor Wavelet Transform and Complex Directional Filter Bank. The conclusion made from this paper is Gabor Wavelet Transform seems to be the most efficient way to do features

extraction since by using this method, the highest image retrieval is accomplished compared to other methods. However, this method has its disadvantages such as its difficulties in computing due to its complicated calculation. Other methods do have their own advantages that can be chosen depending on the process situation and condition of the input images.

Gray Level Co-Occurrence Matrix method provides incredibly good result of features extraction of image with up to 90% of extraction. This method uses graycomatrix function that will calculated the ratio of intensity of image to a pixel. Second method originated from co-occurrence probability obtained will be used to produce texture characteristics in the image [7].

GLCM is also used in [11] to differentiate breast cancer tumor named as benign and malignant. In this paper, the experiment is done in several angles and distance. The sum of the readings is then calculated and entered into a confusion matrix that will be used in K-means classifying method.

2.2.2 Region-of-Interest Technique

Region of Interest is a section in an image that will be chose according to its valuable numbers of elements that can be used for analysis [12]. For example, in an image, only certain part of image is chosen to be extracted depending on numbers of elements available in the specific region or in simpler way of saying, cropping the image to certain part of the cells exist.

Regularization method is one of the methods used to extract some region in an input image. In this method, input image will be reorganized and iterated to acquire a matrix that will be used for calculation to find suitable algorithm. By using this method, better isolation from other region is achieved although its computation complexity is reduced [13].

Next method is Hierarchical Region Processing which will isolate and bound edge of element in an input image from its surrounding. Attenuation difference between the element and background will result in discontinuity between a region to another. However, to ensure better extraction result, image enhancement need to be done in the first place or every time separation is done. Precise and significant result will be achieved after completing this method [14].

Using wavelet method for Region of Interest image processing, image can be divided into several parts and can be represented in wavelet form. Input image will be decomposed into several levels according to the algorithms of wavelet. After performing this method, region that is not important in input image will grow while the subject in region of interest will be well-preserved [12].

2.3 Classification

Classification is needed in the last stage of Image Signal Processing to differentiate between one types of image to another. In this project, classification is needed to differentiate between Nucleolar from Centromere pattern and vice versa. Research has been done to determine what type of classifier is suitable to be used together with texture feature extraction, available in MATLAB and has highest accuracy.

Support Vector Machine or SVM will determine a hyperplane separating one types to another and uses support vectors or boundary pixels to create a decision surface where the testing image will be classified to which type in the program [15]. In SVM, training images will be selected in which the value of differentiating properties will be stated to MATLAB and the group of the pattern is also specified. SVM classifier will then distinguish between the two groups, which one has the nearest value to the testing image [16]. Based on research done, SVM has highest accuracy compared to other alternatives available. SVM is also design for binary classification which is differentiating 2 types of groups only [17].

Maximum-likelihood classification or MLC is a method to estimate parameters of a statistical model in which a data set is applied and MLC will provide the testing image's parameters [18]. MLC will estimate mean and variance of the tested image according to the determined value beforehand. However, it is proven that MLC has a lower accuracy compared to other classifier. This is because when using MLC, intensity takes a very important role in it. Images with low intensity might give wrong results. Therefore, the information is wrong and not suitable to be used in this project as the results might differ and the accuracy will decrease [19]. MLC might be able to classify correctly if the image is in RGB color space [20] and even after the segmentation, the color remain. In this project, the image will either be in grayscale or in black and fluorescent. Therefore, MLC is not suitable to be used.

CHAPTER 3

METHODOLOGY

3.1 Research Methodology

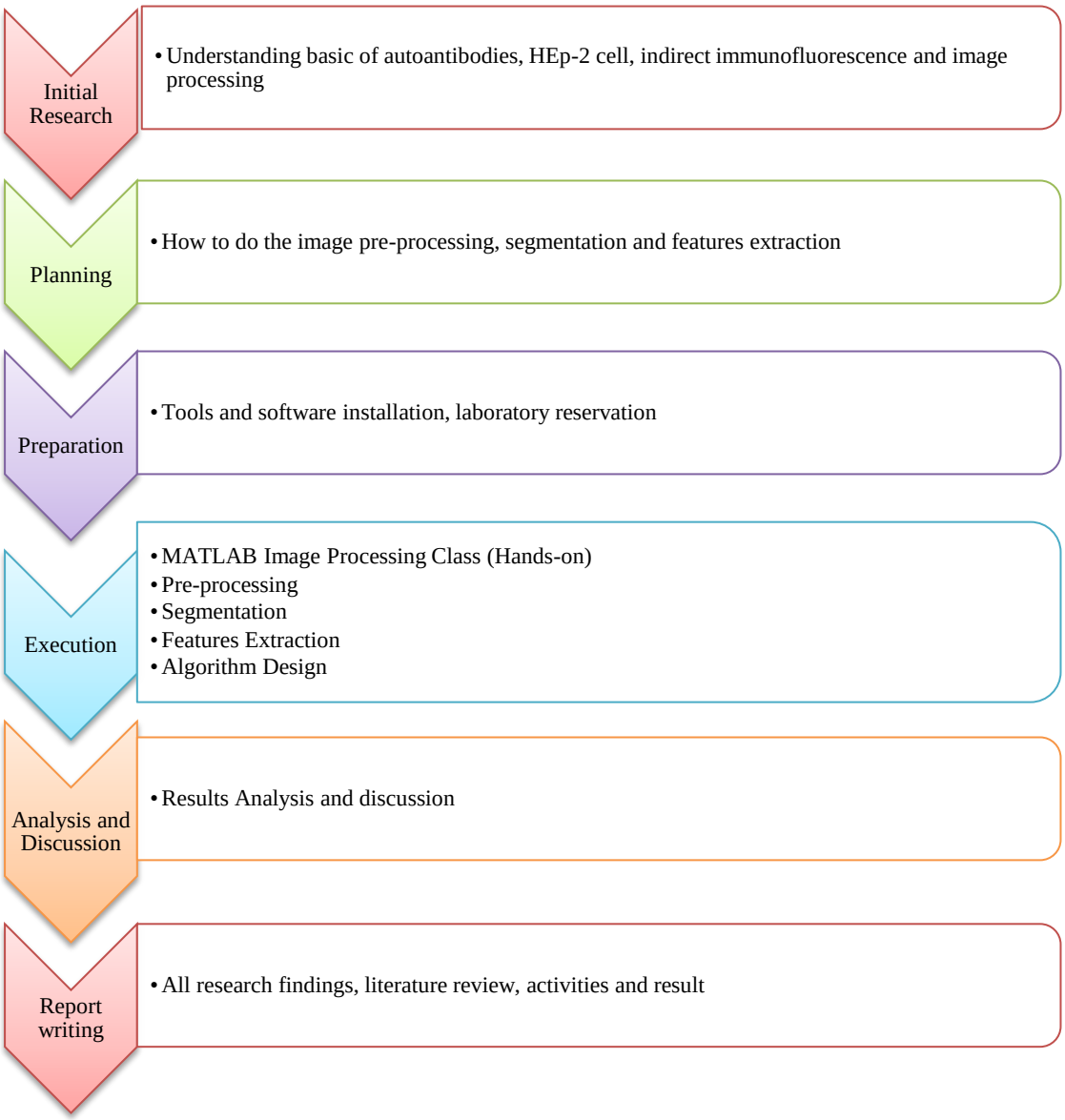

Figure 2: Research Methodology

3.2 Tools

- MATLAB software
- Image Processing Tools in MATLAB
- Graphical User Interface in MATLAB

3.3 Project Activities

Pre-processing is done to obtain clear image of the cells that will be enhanced in segmentation process. The image output of segmentation will be extracted to gain the features of the internal cell. Results for differences between Nucleolar and Centromere have been acquired and will be used to design the algorithm that will identify the patterns once the image is run in MATLAB. Region-of-interest method was also used and combined with feature extraction where the image is cropped to certain parts of the whole image in which the feature of the cell can be clearly seen.

3.3.1 Image Pre-Processing

Pre-processing is performed on input images to reduce noise and to improve image quality such as contrast, brightness and intensity. There are several ways to do it in MATLAB with different functions.

3.3.1.1 Color Space Conversion

Color space is color representation of each input image loaded into MATLAB. There is several color spaces available in images however, not all color space are applicable in MATLAB. There are several color spaces that will give inaccurate results in reading the features of images or there are also problems when MATLAB functions cannot identify the color space. Therefore, color space conversion is a very important pre-processing to be done before proceeding in other MATLAB image processing.

The coding in Appendix A shows how color space is converted from RGB which is represented by the original image with red, green and blue color space into other color space such as grayscale, negative and log output image that will be used as input image to other processes.

The result from the coding is as follows:

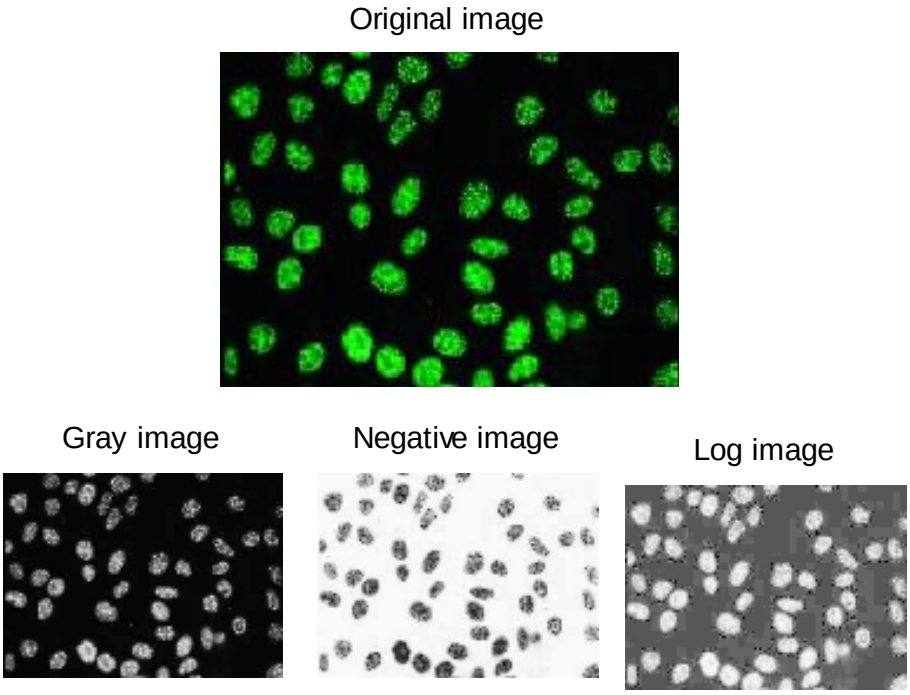

Figure 3: Result of Color Space Conversion

Based on the result obtained from the color space conversion pre-processing, all three types of color space can be used in this project as the features are well-preserved and the background can be separated from the cells itself. Therefore, further studies and experiment need to be held to determine which one is the most suitable color space to be used with the algorithm later on.

3.3.1.2 Histogram Processing

In order to do histogram equalization, image will first need to be converted into grayscale as histogram can only display for binary value which in this case will be black and white. To convert image to grayscale, type of input image need to be defined first. For this example, RGB (Red, Green and Blue) image was used.

Appendix B illustrates the coding used to execute histogram processing in which the image will be converted into grayscale from RGB and its histogram will be displayed accordingly. The histogram of the grayscale image is then stretched and equalized and the result of the edited input image is as displayed in Figure 4.

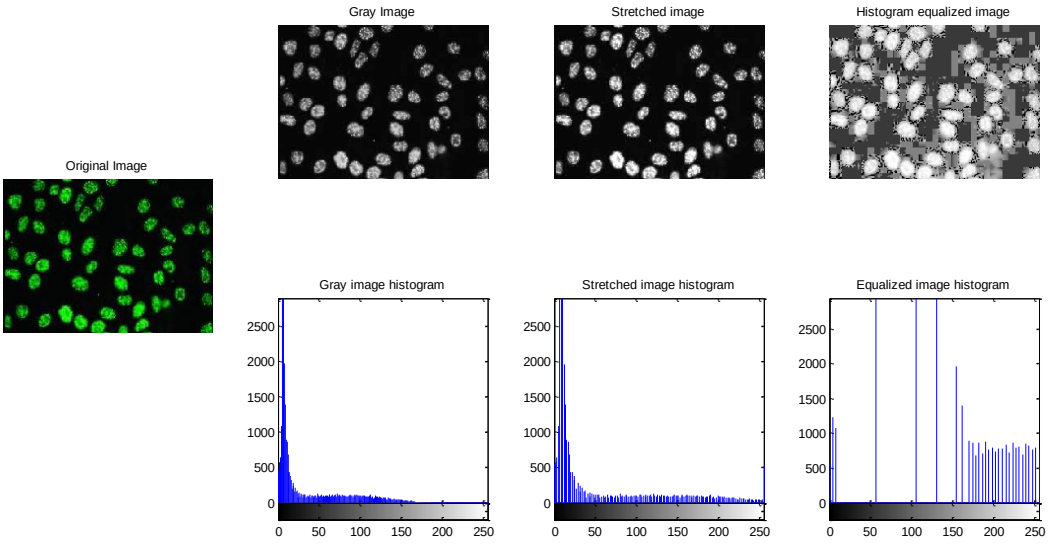

Figure 4: Histogram Processing Result

Figure 4 shows how original RGB image is converted to grayscale and the grayscale image histogram. The histogram is adjusted to stretch from value of 0 to 255 and the output image is seems to be clearer, no noise and better contrast. The grayscale image histogram is then equalized. However, the output image is not clear as there are noise, low contrast and unclear output image. Therefore, histogram equalization is not suitable for this input image and stretched image appears to be the best result to be used for further analysis on this image.

3.3.1.3 Noise Reduction

Noise can cause inaccuracy when doing analysis of image by using MATLAB processing where the properties of the images can be read wrongly and the features of the image will be also wrong. Therefore, to overcome this problem, noise need to be reduced and if possible, eliminated if the correct functions from MATLAB image processing tools are used. There are several functions to reduce noise available in MATLAB and those functions need to be chosen based on types of noise that need to be reduced.

Appendix C shows MATLAB coding on how noise is first added to original image to introduce different types of noise available in MATLAB. After recognizing the types of noise, by using naked eyes, we will be able to define types of noise appeared in the input images and appropriate filter can be used to reduce the noise. The result of the above coding is shown in Figure 5.

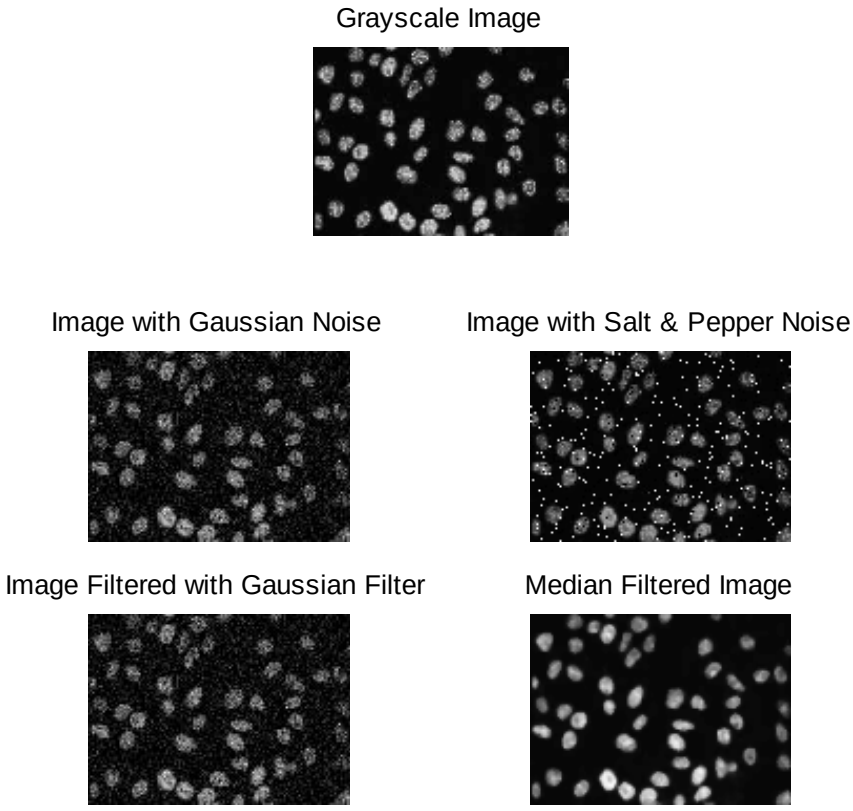

Figure 5: Result of Noise Reduction

From Figure 5, it is concluded that different types of noise require suitable types of filter to overcome it. From this coding, Gaussian and Salt & Pepper noise is introduced. For input image with added Gaussian noise, Gaussian filter is used to eliminate the noise. However, it is seen that the noise is not removed nor reduced. A conclusion can be made that Gaussian Filter is not suitable for Gaussian noise. The second noise added is Salt & Pepper noise and Median Filter is used to eliminate the noise. The output image shows that the noise has been removed and the output image looks like an image without noise. This can be concluded that Median filter is suitable to be used with Salt & Pepper noise.

After knowing the types of noise, suitable filter will be used and in which the image will be cleared out and become clearer. If the filter is not suitable, the image will remain with noise and unable to be used for analysis.

3.3.1.4 Image Sharpen Process

Input image to be used for segmentation needs to be sharpened so that better output image can be obtained and used in feature extraction. Sharpening the images

will also sharpen the features of the image hence making the characteristics of patterns clearer.

The coding in Appendix D is used to sharpen the cropped and resized input images that will give better images. From the coding, the original image is cropped in Figure 1 of the subplot, the original image by dragging the mouse to form a box shape and double click the box to select the cell. The cropped image is then resized to 256 by 256 pixels and sharpens to make the image clearer.

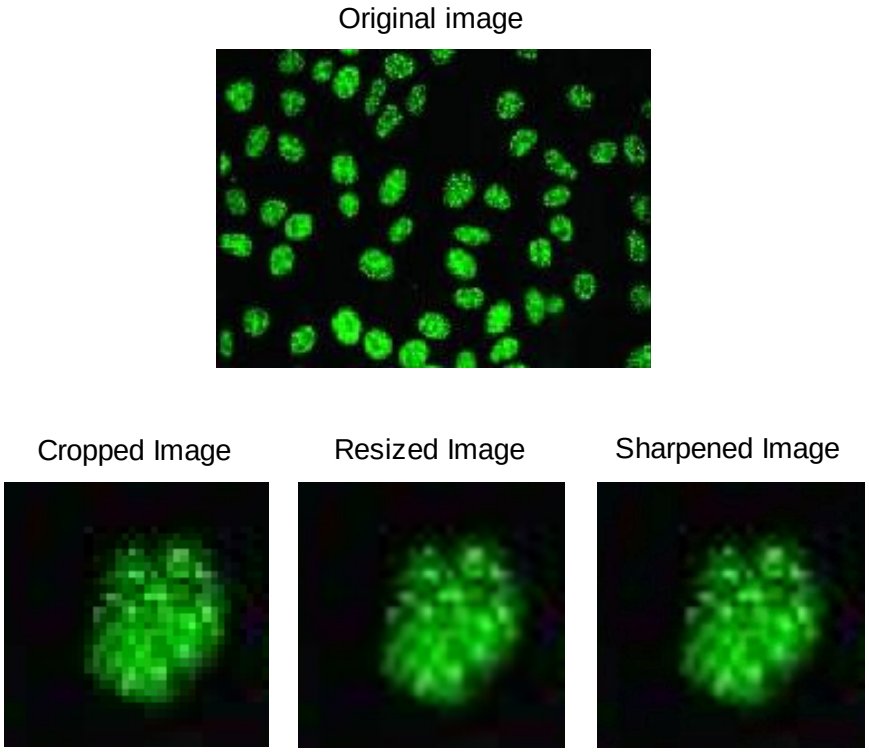

Figure 6: Output Image for Sharpened Image

The result above shows that the sharpened image gives us clearer features in the cells where the nucleus can be seen more clearly compared to the resized image. Segmentation and other filters will be added so that better image can be seen and features can be extracted with high accuracy.

3.3.2 Image Segmentation

Image segmentation is a very important part in image processing where result from segmentation process will be used for features extraction and classification stage later on. In segmentation, image is enhanced according to edge of its elements in input image or combined elements will be separated from each other. There are several types of image segmentation such as threshold segmentation and edge detection.

Threshold is the simplest way to do segmentation. Threshold can be divided into several divisions such as histogram shape-based and entropy. In histogram shape-based, histogram will be analyzed and smoothen. Threshold segmentation is said to be the most stable method for image segmentation [21].

In edge detection, intensity will be the main parameter to differentiate element to the surrounding background. Edge detection has a very close relationship with region of interest. In order to be able to segment an object, a closed region borders is needed.

However, for this project, texture segmentation is the best approach to enhance the images as intensity of the cells is different from its surrounding. By using texture segmentation, nucleus can be well-preserved and the surrounding can still be seen. If other approach such as edge detection is used, the cell will appear in black and white and the property of ANA pattern will be lost.

The steps to do texture segmentation are as follows:

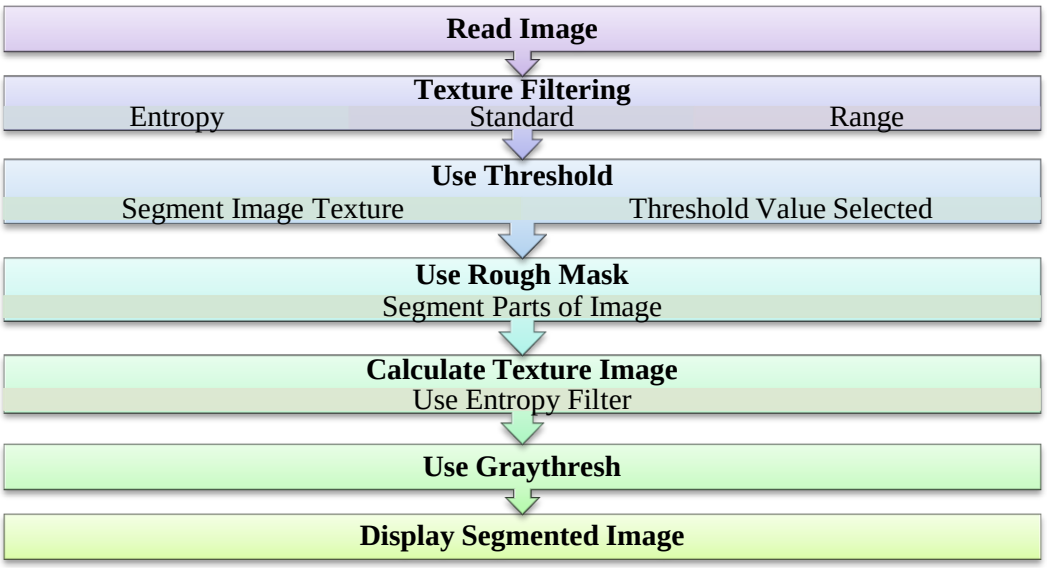

Figure 7: Texture Segmentation Steps

Basic steps to do texture segmentation is first to read the input image. Depending on the types of filter to be used, some preprocessing need to be done such as converting RGB to grayscale meanwhile some filtering do not require that. Texture filtering will be done to the input image either in RGB or grayscale. The three types of texture filters that will be used in this project which is entropy, standard and range. These three filters will produce significantly different results in image and also calculation. After the filtering is done, the image will then undergo threshold depending on the suitable value of threshold for the image. This is to enhance the image to become clearer. Up to this point, this project has completed the threshold step.

Later on, texture image will be calculated by using entropy filtering. However, further studies will be done to see whether or not there is any other approach to calculate this image. After calculation has been done, graythresh function will be applied to the image to obtain better result of image and lastly, the final segmented image will be displayed and analyzed to be further used in feature extraction.

However, for this project, threshold is not suitable as over-segmentation on the cells will happen. Therefore, other segmentation process had been done to test its suitability with the objective of this project.

3.3.2.1 Texture Filtering

Texture filtering is a method to obtain the texture color of the texture mapped pixels in an image. There are several filters that can be used for texture filtering such as entropy filter, standard filter and range filter. To obtain the best result, all three filters are used with both RGB and grayscale input images.

3.3.2.1.1 Entropy Filter

Entropy filter allows output pixel with entropy value of 9 by 9 neighborhoods corresponding to pixels of input image. The first result is by using RGB input image where the original image is used. It shows that entropy filter did not produce a good result compared to other two filters.

In the coding at Appendix E-1, the input is first read from the stored location in the computer. Then, the image is shown in figure at subplot with 2 rows, 1 column. Entropy filter is then applied to the image and mat2gray function is used to rescale the filtered image. The rescaled image is then shown in the first picture of second row in

the figure. Threshold with value of 0.8 is used to the filtered image to segment the texture. Value of 0.8 is chosen because it more or less the intensity value of pixels along the boundary of the texture in input image.

The result of entropy filtering is as shown.

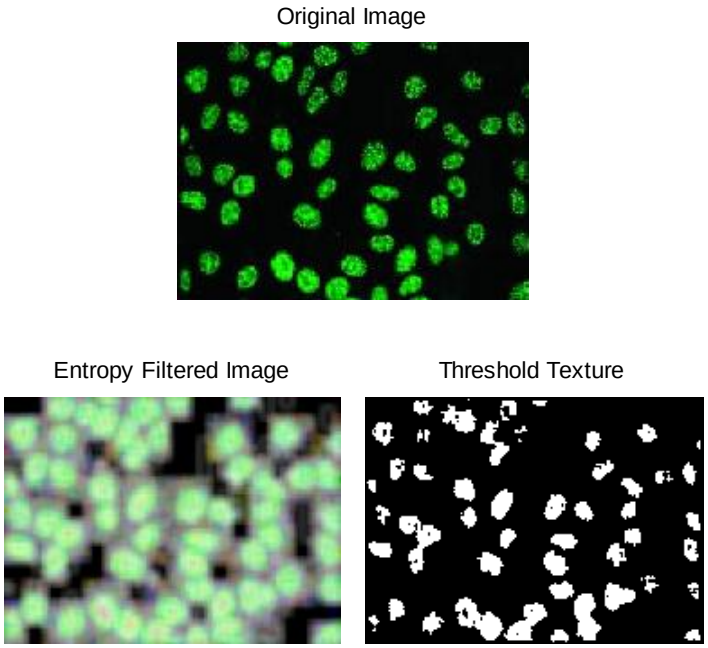

Figure 8: Entropy Filter for RGB Input Image

Based on the result, it is shown that the both segmentation is not suitable for this project as the features for all cells in the input image has lost and the results will cause inaccurate reading. Over-segmentation in Threshold Texture causing all the features properties in the cells are missing. However, further studies has been done to obtain the result of grayscale image and later on to be compared with RGB input image result.

The MATLAB coding for grayscale input image is different from the previous coding as it converted RGB image to grayscale first as attached in Appendix E-2. Then, the grayscale image is used in entropy filtering. The result is then proceeds with mat2gray function just like previous coding and then is added with threshold value of 0.8. The result of the above coding is as attached.

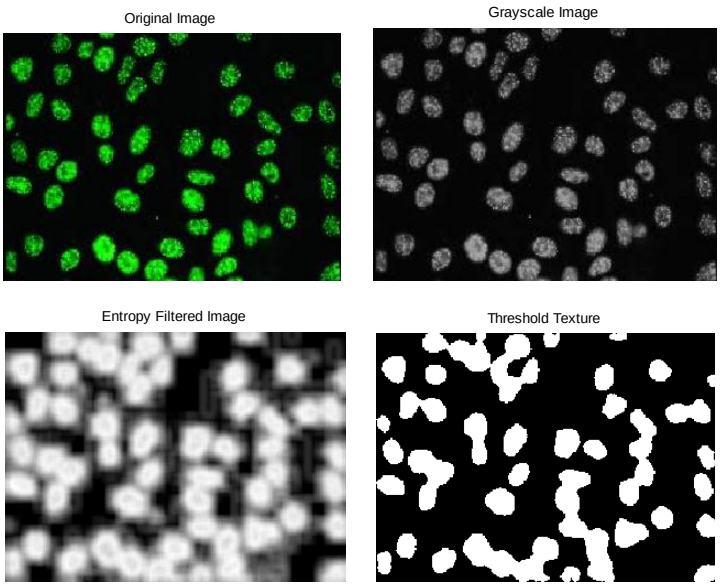

Figure 9: Entropy Filter for Grayscale Input Image

Based on the result obtained, it can be seen that the threshold texture image is not clear compared to the RGB input image result. This is because by converting RGB image to grayscale image, the texture of the image has blended together. However, grayscale image is much clearer as compared to RGB image. Therefore, it can be concluded that both segmentation is not suitable for this project.

3.3.2.1.2 Standard Filtering

Standard filtering is a function in which the output pixels contain standard deviation value of 3 by 3 neighborhoods in corresponding of input image. For this filter, both RGB and grayscale input images are used. The differences of both results are then compared to see what type of input image works for this filter.

For MATLAB coding in Appendix F-1, RGB image is used. The basic coding which is to read image and displaying the image in figure is just the same as entropy filtering coding before. Stdfilt function is used, followed by mat2gray function which is to convert matrix into grayscale image and the filtered image is showed in the figure.

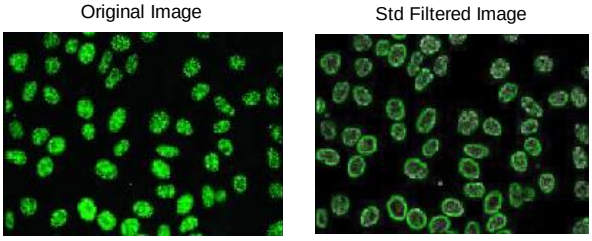

Figure 10: Standard Filter for RGB Input Image

The result shows that after undergo standard filtering; the edge of each cell can be seen clearly and better features preservation take place as compared to Threshold Texture.

The coding is then redone with grayscale input image to see the differences. Please refer to Appendix F-2. In this MATLAB Coding, the input image is first converted into grayscale image and later on used in the standard filter.

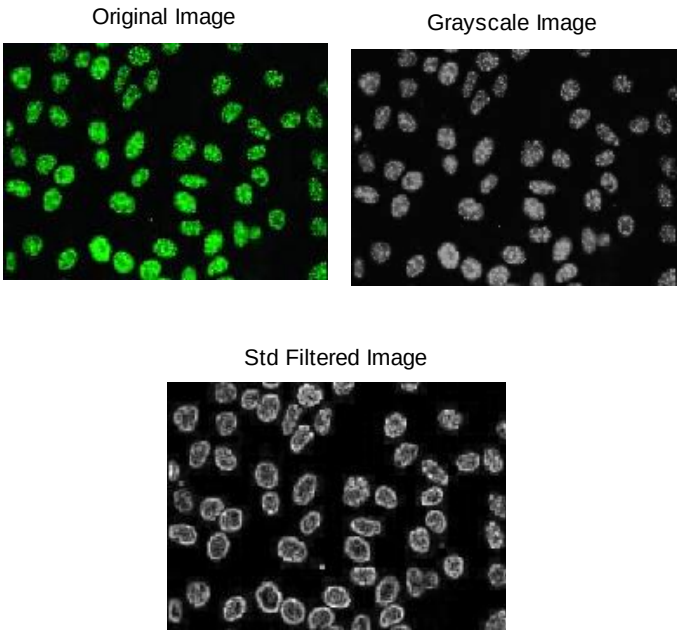

Figure 11: Standard Filter for Grayscale Input Image

The result obtained from grayscale input image better than RGB input image result as clearer features in every cell can be seen. This can be concluded that grayscale is better to be used with Standard Filter.

3.3.2.1.3 Range Filtering

Range filtering is almost the same as standard filtering where the neighborhood of 3 by 3 is calculated for the input image. However, by using range filtering, range value such as minimum to maximum value of pixels in output image can be obtained.

For this filtering, both RGB and grayscale input images are used to see the difference of results where RGB input image is used to be filtered by using rangefilt function. The result from coding in Appendix G-1 is as shown below.

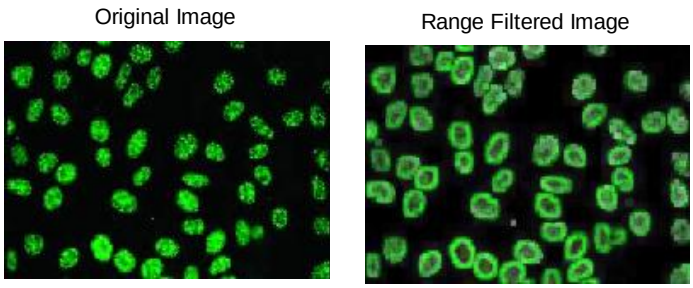

Figure 12: Range Filter for RGB Input Image

The result obtained is almost similar to standard filtering result. This is because both filtering are in 3 by 3 neighborhood. Therefore, the calculation is also nearly the same.

Next, MATLAB coding is done for grayscale input image such as in Appendix G-2. Same as previous filtering, the input image is first converted into grayscale image and then used in range filtering to obtain the result which then been threshold with value of 0.8.

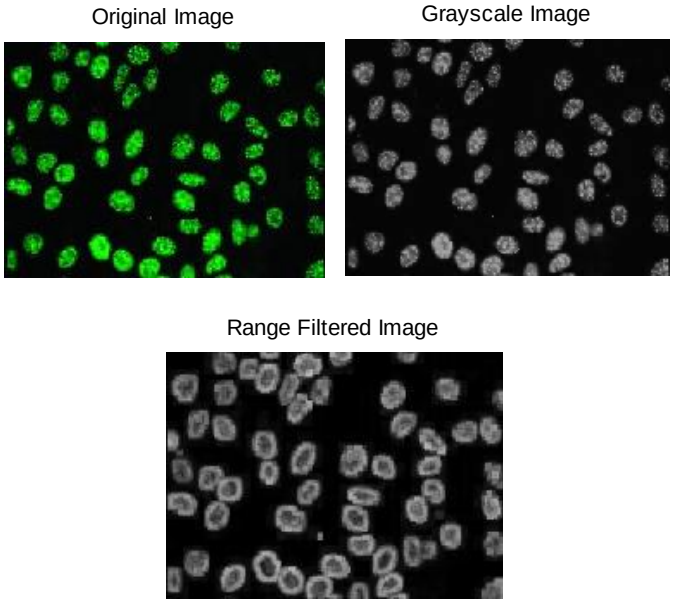

Figure 13: Range Filter for Grayscale Image

The result obtained is similar to the result for Standard Filter for Grayscale image. This is because range filter is similar for Standard Filter by using 3 by 3 neighboring.

Out of these three texture filters for texture segmentation that has been used in this project, it is concluded that Standard Filtering and Range Filtering can be used in this project while Threshold Texture filtering cannot be used although it can

differentiate the texture of cells clearly. This is because, in this project, the features of the project are required to be well-preserved and as clear as possible.

The color space that will be used is grayscale as grayscale shows the most accurate data compared to other color space and most of MATLAB functions are compatible with grayscale color space.

3.3.2.1.4 Gradient Magnitude

Gradient magnitude is for identifying the edge of image and discontinues the strength of the intensity accordingly. There are several operators that can be used in the the program and Sobel is the most common choice. In this project, the cells and nucleus's edge is clarified by using gradient magnitude to obtain clear internal image of the cell. The MATLAB coding is as shown in Appendix H.

In the coding, Sobel operator is used the function is differentiated. The image is then filtered by using the original and differentiated function. Then, the image is combined together to perform a new image. The result of the coding is as below.

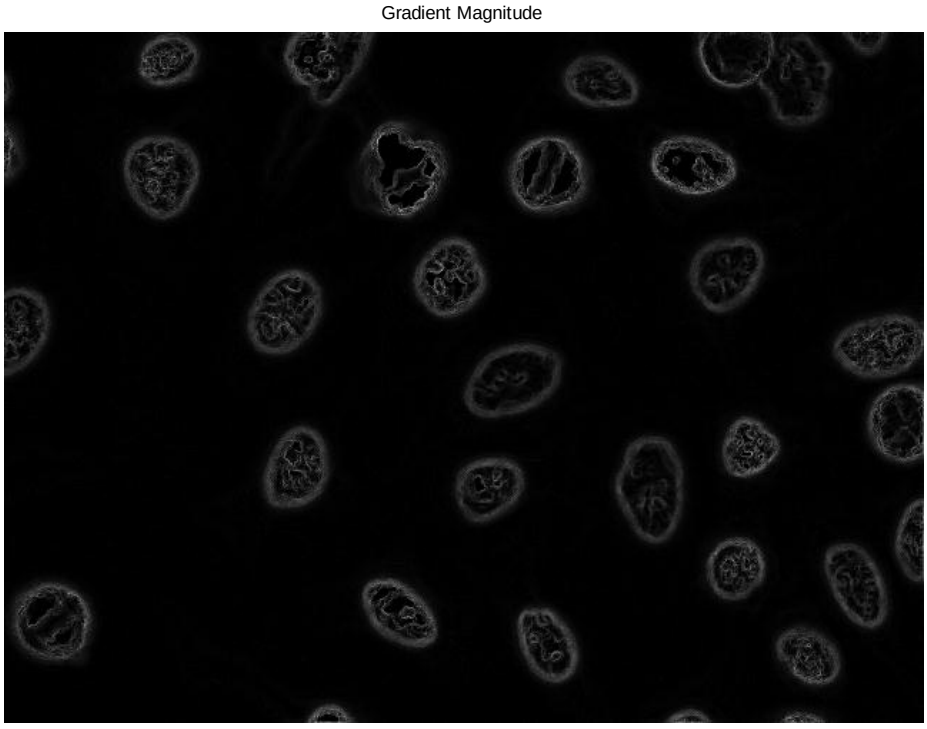

Figure 14: Gradient Magnitude Result for Centromere

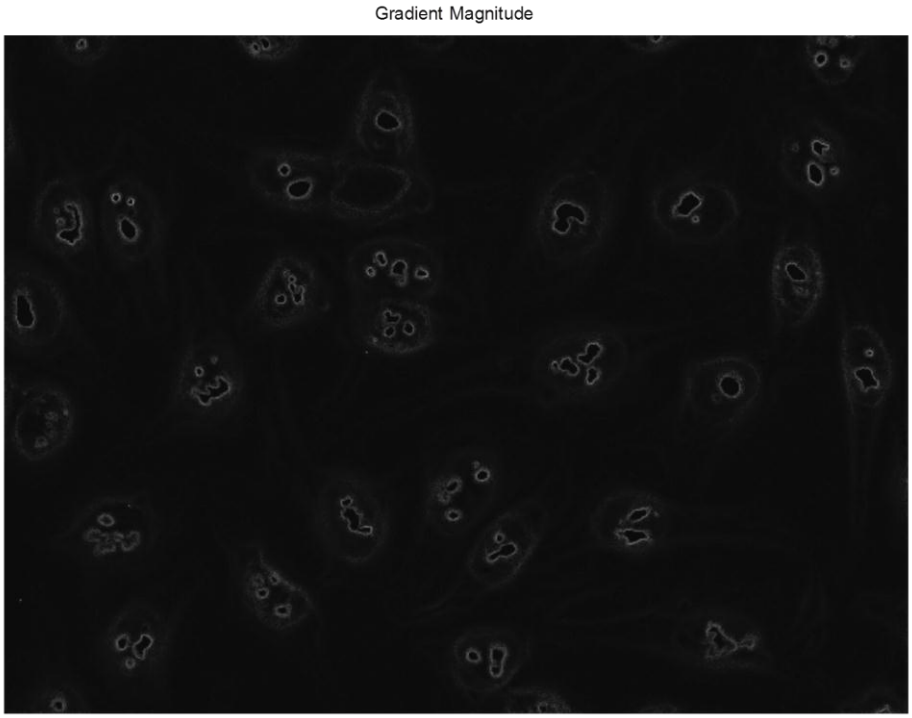

Figure 15: Gradient Magnitude Result for Nucleolar

Based on the results above, it is seen that the segmented image of Centromere and Nucleolar produces clear nucleus of the internal cell that will be very important later on to be used in features extraction and classification. Therefore, gradient magnitude is chosen to be used in the program in this project.

3.3.3 Features Extraction

Feature extraction that will be used in this project is divided into two techniques which are texture feature extraction and region-of-interest. These two techniques are combined to ensure that the cell in every image can be seen clearly and better feature properties reading can be obtained.

3.3.3.1 Region – of –Interest

For some cases, input images need to be cropped and resize so that certain parts in the input image can be selected, enhanced and can be clearly chosen as the main element for the experiment. In this project, the input image is cropped to only one cell so that features for the cell can be clearly seen and accurate reading on the cell's features can be obtained.

The MATLAB coding for the cropping process is as attached in Appendix I. From the coding, the input image is defined and the image is cropped at selected rectangular selection. After adjusting the position and size of the cropping area, double-clicking in the box will automatically crop the cell. For every image, the position is different depending on the coordinate of the image need to be cropped. The output image is then displayed side by side with the original image.

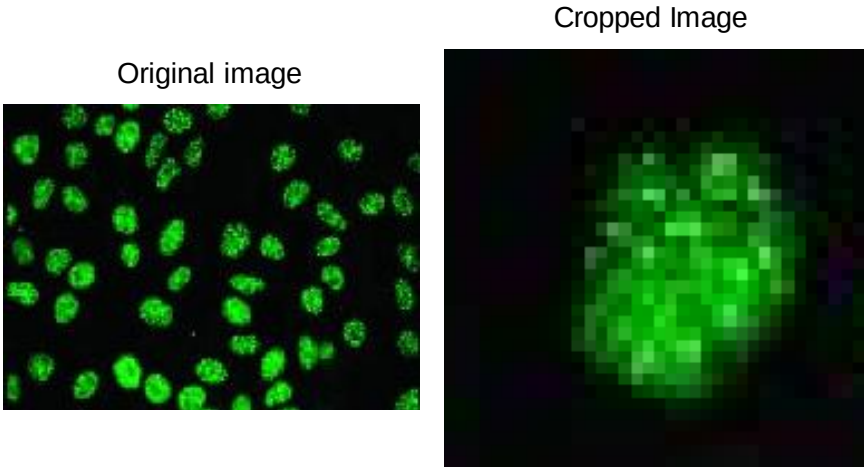

Figure 16: Output Image of Cropping Process

Based on the result above, the cropped output image is unclear and blurred. This shows that further pre-process and segmentation need to be done so that the features of the cell can be seen clearly and the characteristics of the image can be obtained for further steps in the project.

3.3.3.2 Texture Technique

Texture feature extractions require the reading of cell properties to differentiate from one pattern to another. In this project, texture feature extraction that is used is Gray Level Co Occurrence Matrix of also called as GLCM. Among other feature extraction available, GLCM has high accuracy which is up to 99%. This accuracy can be increased in this project as region-of-interest is selected to include only one cell for the process. Therefore, the properties reading will only focus to the internal part of the cell which is its nucleus. For Centromere pattern, number of nucleus is more compared to Nucleolar and the position of nucleus in each cell is closer to each other compared to Nucleolar.

MATLAB Coding used for GLCM is as in Appendix H. Graycomatrix function will show an 8 by 8 matrix of properties for the cell cropped in the algorithm. To ease the reading of result from the matrix, graycoprops function is used to view the main properties of the image together with its minimum and maximum reading. The properties that will be shown are Energy, Homogeneity, Contrast and Correlation.

Analysis is done to determine which property has the highest difference result reading between the two patterns (refer to Chapter 4 for Result). After several images have been tested, it is seen that Energy has the highest difference; therefore, Energy is chosen to be the property that will classify Nucleolar from Centromere. However, further testing on more images will be done to ensure that Energy is the most suitable property to be used throughout the project.

3.3.4 Classification

Support Vector Machine or SVM will be used to classify the cell patterns in this project. SVM requires pre-determined value of the cells in the coding before it is able to classify the tested image. In this project, 118 training cells are used and the energy values of the cells are specified in the coding according to GLCM results. Later on, the tested image will be differentiated according to which pattern hyperplane it is nearer to.

The MATLAB coding of the classification is as attached in Appendix K. From the MATLAB coding, the Energy value of each cell has been set according to GLCM results in the feature extraction. After the values are mentioned, the training image is defined and the pattern groups of the training image is clarified to be either 0 or 1 where 0 is for Centromere pattern and 1 is for Nucleolar pattern. In the end, the tested image will be classified according to the pre-determined group and Energy value and the result of the pattern's name will be observed at the command window.

3.3.5 Graphical User Interface

Graphical user interface or GUI is a type of interface that will let users to communicate with the program by using graphical icons and visual indicators instead of typing text commands.

In MATLAB, GUI can be created by typing ‘guide’ in the command window. A pop-up window will appear asking to select either blank GUI or programmed GUI. In this project, Blank GUI is used and the axes, push buttons and edit texts are arranged accordingly and in a manner at which users can easily understand the flow.

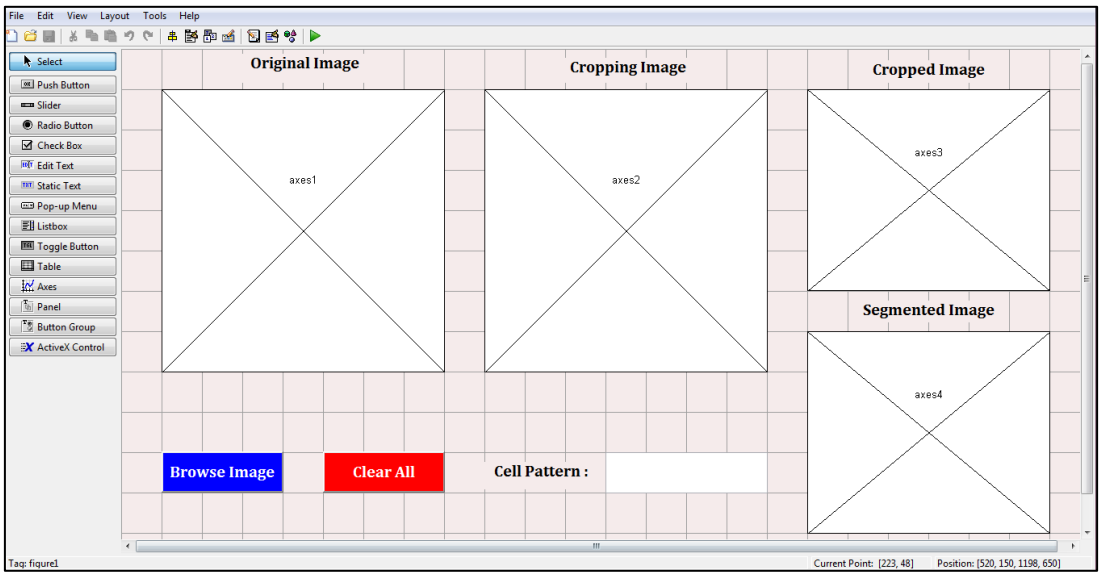

Figure 17: GUI Creating Steps

After the arrangement has been done, the GUI figure is saved and an editor consisting of the handles in the GUI will appear. Program coding of the necessary program is then typed into the editor according to its steps. To show image in the GUI, instead of using function ‘*figure(1);*’ or ‘*subplot (1,2,2);*’, a function of ‘*handles.axes1*’ is used and the image will be displayed on the first axes in the GUI.

By using GUI, users can easily run the program without the necessity of exposing the coding to the users. This will also minimize the time taken by the users to run and will avoid the probability of the coding being unconsciously modified by the users.

3.3.6 Key Milestone

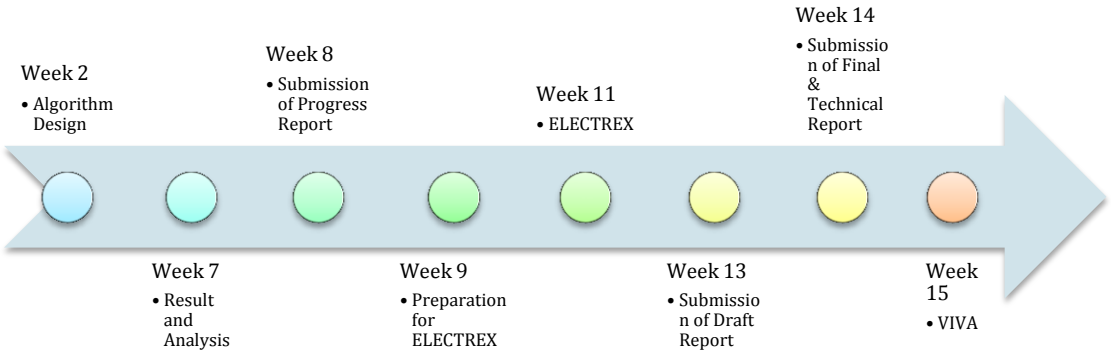

Figure 18: Key Milestone

As per shown in the above figure, in Week 2, an algorithm that will automatically define ANA pattern to either Nucleolar or Centromere will be designed and it is estimated to take place in about five week. In Week 7, after the results have been obtained, an analysis will be done to determine whether or not the algorithm can be used in hospital to differentiate ANA patterns. If there are any problems, enhancement will be done and discussion with supervisor will be conducted for suggestions and changes will be done.

Progress report will be submitted in Week 8 to Turn-it-in website which per required by Final Year Project 2 course. This website will detect plagiarisms and will notify lecturers in charge.

ELECTREX or also known as Pre-Sedex for Electrical and Electronics Engineering student will be held on the 4th of December 2013 which is in Week 11. Therefore, posters and coding need to be completed beforehand to ensure the project presentation will be done smoothly. In order to complete preparation beforehand, preparations will be done starting from Week 9.

Hard copy draft of Final and Technical Report will be submitted to respective supervisor a week before the real submission date which is in Week 14. Lastly, VIVA will be held on Week 15, which is in study week of the semester.

3.3.7 Gantt Chart

Please refer to Appendix M.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Features Extraction

After features extraction has been completed, the property's values of the internal cell can be obtained. Out of the four properties obtained which is energy, correlation, homogeneity and contrast, the best property with different value between Nucleolar and Centromere pattern is chosen as the training value in SVM classification later on. Training cells selected are 118 cells where 76 is Centromere pattern while another 42 cells are Nucleolar pattern. The results are tabulated in table attached in Appendix L.

Based on the table in Appendix L, graphs are drawn to determine out of these four properties (Energy, Homogeneity, Contrast and Correlation), which one of them does not have any overlap between the two patterns. Based on the graphs, the property value that will be used in the classifier will be decided and the tested image will be classified by using the property.

The first property obtained by using GLCM feature extraction is Energy. Energy shows how gray levels are distributed in an image. The higher the number of gray levels, the lower the energy is.

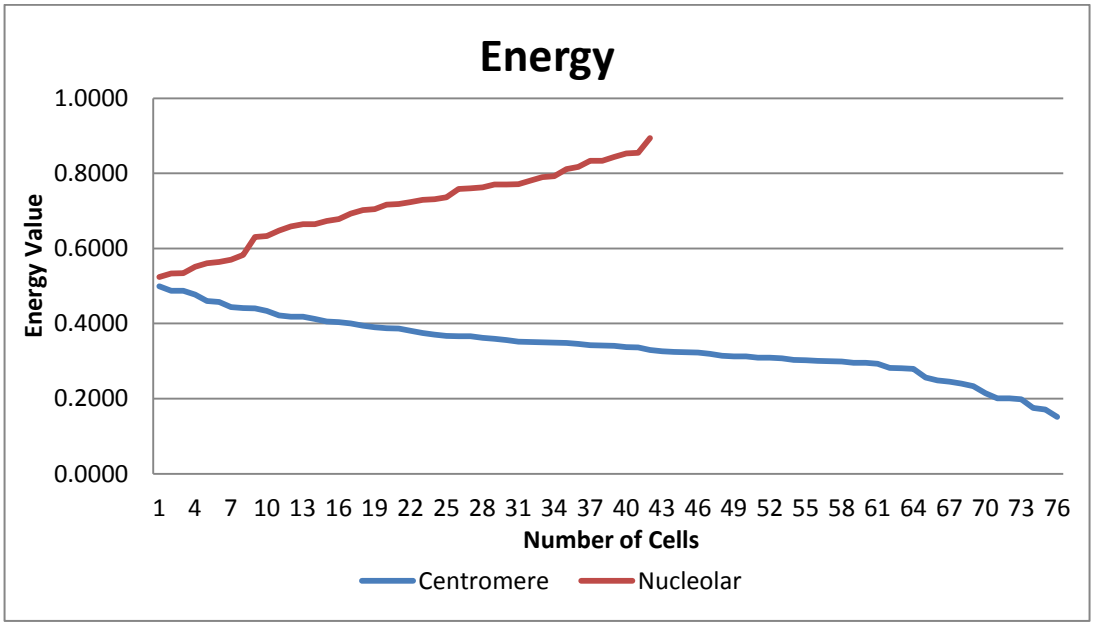

Figure 19: Energy Property Graph

Based on the graph acquired from the GLCM, the energy value of Centromere pattern is lower than 0.5 which is lower compared to energy value of Nucleolar with value of higher than 0.52. This is because, after the segmentation process, Centromere cell will have more gray levels at the edge of the cell and also at the nucleus while for Nucleolar cell; there will be no cell edge that can be viewed only clear edge of cell nucleus. This is why Nucleolar has higher energy value compared to the Centromere pattern. Based on these results, energy property can be used to differentiate between a pattern to another.

The second graph that can be plotted is based on homogeneity. Homogeneity is a way to determine the coordinates of a point in which is defined in a function. Homogeneity will define whether or not the structure in an image of having identical cumulative of function or values.

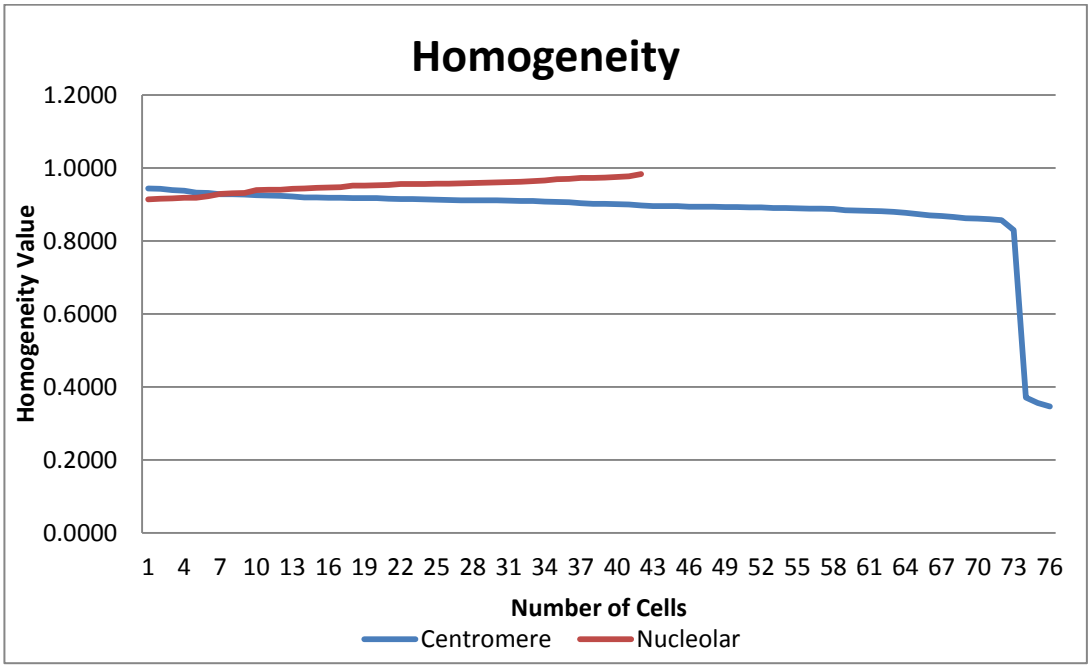

Figure 20: Homogeneity Property Graph

Based on the graph above, the homogeneity of the two patterns are overlapping each other with value between 0.91 to 0.94. For Centromere, the homogeneity value varies from 0.35 to 0.94 while for Nucleolar is from 0.91 to 0.98. This shows that both patterns have nearly the same structure in image and this property cannot be used to differentiate between the two patterns.

The third property that is obtained from GLCM is contrast. Contrast measure the difference in luminance or color that is represented by the image. Contrast can also be said that the difference in color and brightness between the foreground and

background. In this project, contrast obtained is between the background of the cell which is represented by black and the foreground which is the cell represented by gray level.

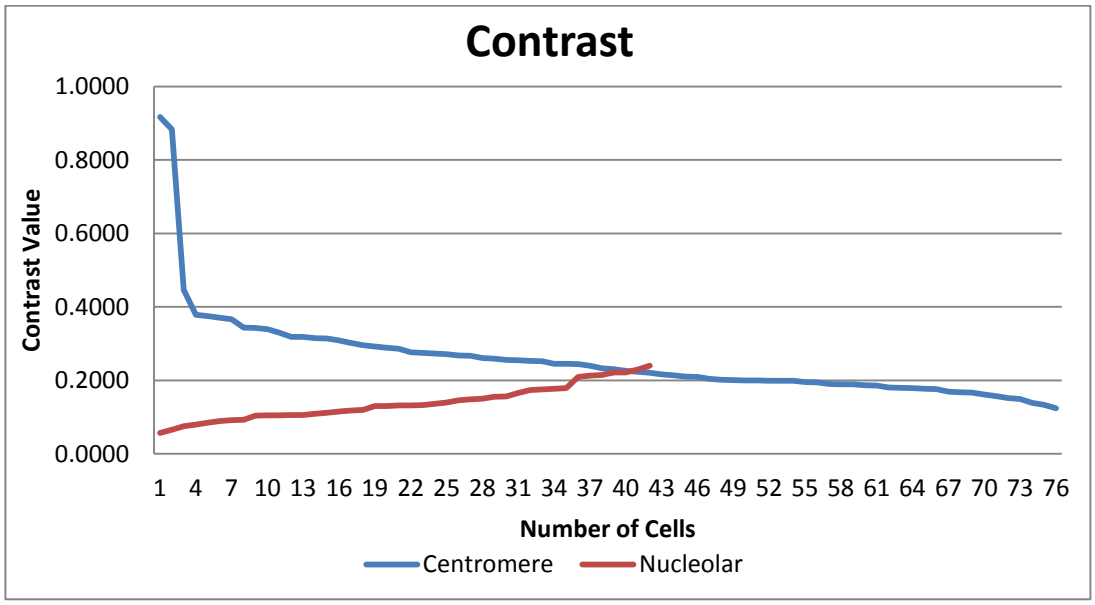

Figure 21: Contrast Property Graph

The result obtained shows that contrast value of Centromere ranged from 0.12 to 0.92 where overlap of the two patterns occurred because Nucleolar has contrast property of 0.06 to 0.24. This is because contrasts of both patterns are almost the same where both patterns are has black background and gray level foreground. Therefore, this property cannot be used to classify the patterns later on.

The last property acquired from GLCM is correlation where correlation is the causal or reciprocal relationship of the pixel in the image tested. Correlation is used to measure the changes occurred in the image by tracking the pixels in the image. The higher the correlation, the higher the causality of the image between one pixel to another.

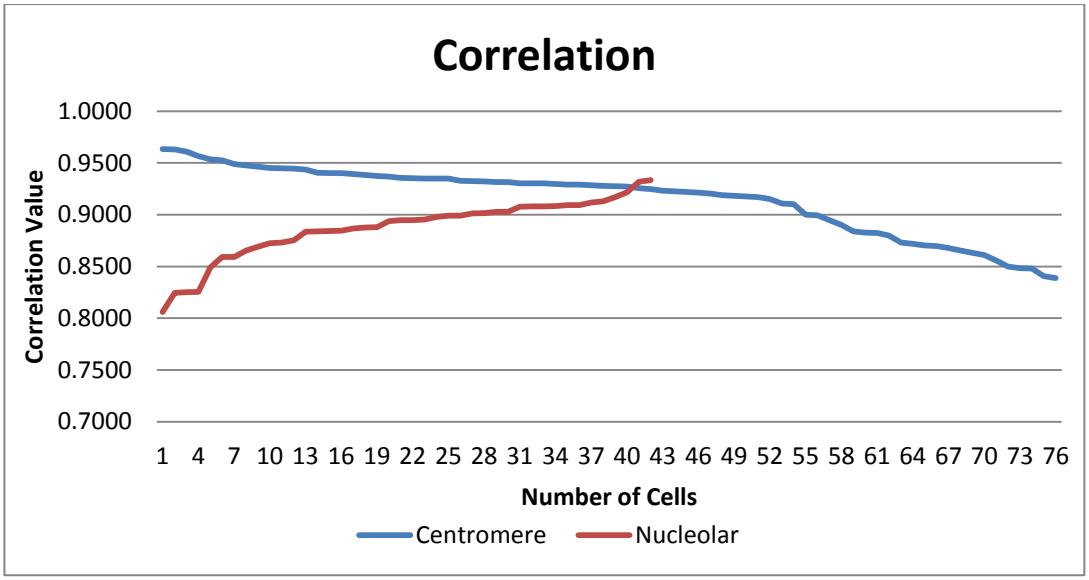

Figure 22: Correlation Property Graph

Based on the graph attained, the correlation values between Centromere are overlapping each other and the values are almost the same. There are no clear differences that can classify Centromere from Nucleolar pattern according to this property. Therefore, correlation is not suitable to be used to classify the pattern in the next stage.

Energy is the only significant feature that can be used to classify Centromere from Nucleolar pattern. Other properties do not have significant difference that can be used to classify.

4.2 Final Coding

The final coding coding for this project is as attached in Appendix N. Based on the coding, the color space of the image is converted into grayscale and pre-processed by using stretch histogram to increase the brightness of the image, wiener filter to reduce the noise of the image and Gaussian filter to smoothen the image. The image is then cropped to select the region-on-interest which the cell of ANA pattern, adjusted the contrast of the cell and segmented using gradient magnitude technique. The features of the cell are then extracted by using Gray Level Co-Occurrence Matrix and classified according to the results by using Support Vector Machine.

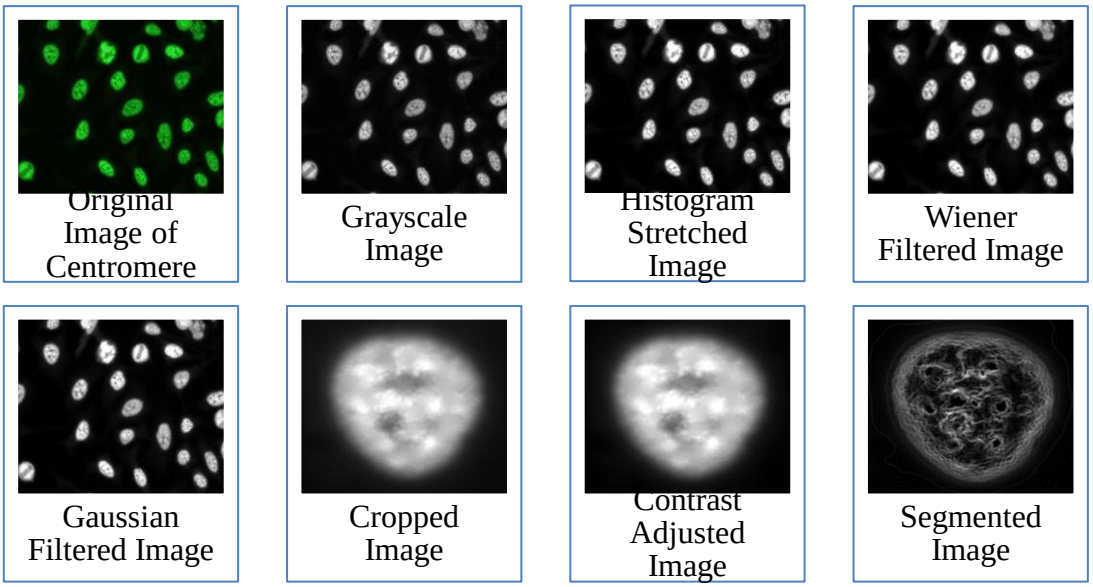

Figure 23: Centromere Final Coding Results

Based on the results above, it can be seen that the internal texture of the cell can be observed and properties values of the internal of the cell can be obtained from GLCM features extraction. Next, Nucleolar pattern processing is done to ensure that the same coding can be used for both pattern and the last results between Centromere and Nucleolar are different from each other.

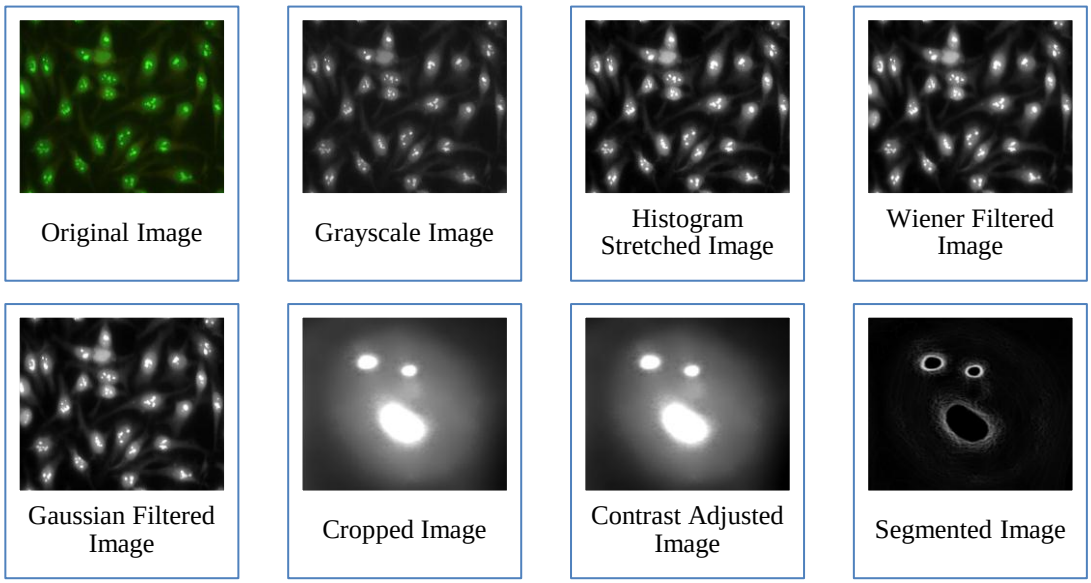

Figure 24: Nucleolar Final Coding Results

Nucleolar processing shows that clear inner nucleus of the cell can be obtained and the final image is different compared to Centromere. Therefore, the features of the internal is different from each other can classification can be done. However, GLCM features extraction is needed to obtain the final reading of properties.

4.3 Graphical User Interface

In order to make the program as a user-friendly program, Graphical User Interface or GUI is used. This interface that uses graphical icon that let the users to easily browse the testing image from the computer storage, cropping the wanted cell and obtained the results at a text box. GUI provides easier access to the program as compared to classical MATLAB programming where the users will have access to the coding which will be disadvantage to the vendor selling the coding.

In the GUI coding, all the coding done stage by stage are combined together to form an algorithm that can be used by the operator in distinguishing ANA pattern later on. Beginning with pre-processing of the image, followed by cropping a cell by choosing the cell wanted to be used and segmenting the image and then the features of the cell is obtained and used to be classified into which pattern. A reset button is included if the operator wish to clear all images and results and load another image to be classified. The final result of the GUI coding is as attached below.

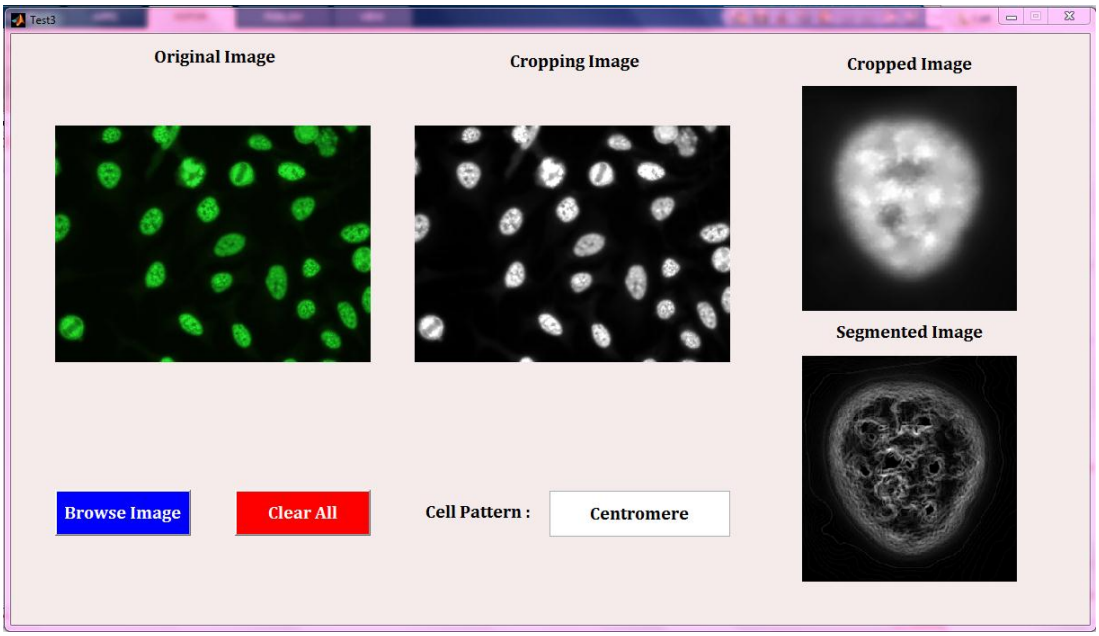

Figure 25: GUI Result of Centromere

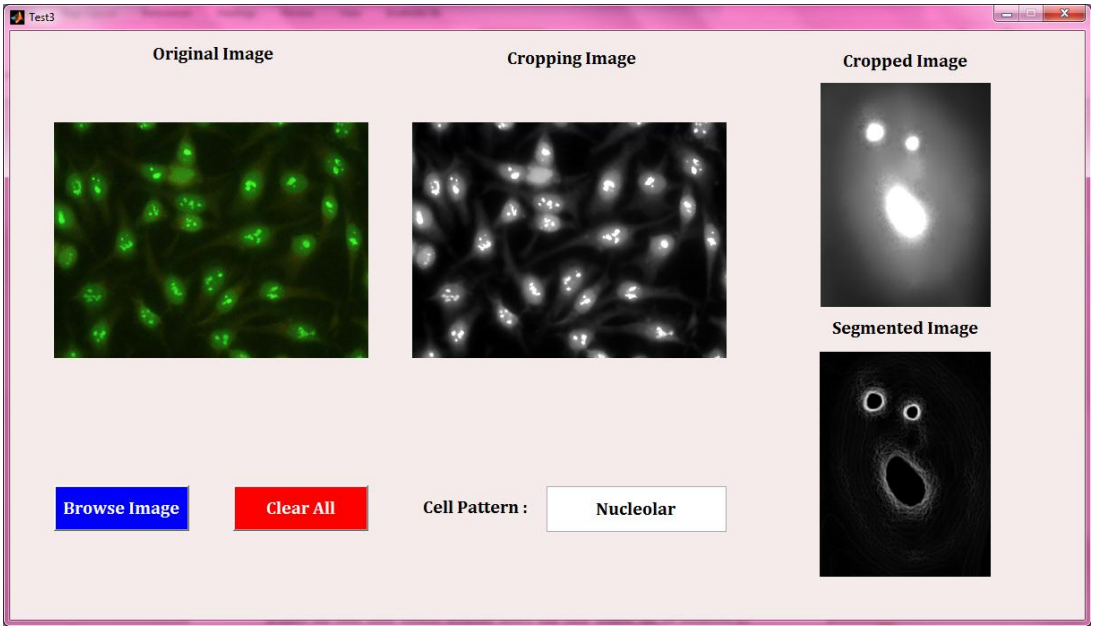

Figure 26: GUI Result for Nucleolar

From the images, clear view of internal nucleus of the cell can be observed and operators will minimize the time taken to differentiate ANA patterns and will have more than 90% of accuracy in classifying the cells compared to the classical way of classifying which is by using their naked eyes.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Results obtained shows that color images can be enhanced by several methods either by converting to grayscale or by using the original images. Output of image segmentation shows sharp images with clear texture difference between the element and background. This has been such a help for further progress in the project which require enhanced and clear images in order to acquire the best result of feature extraction. The steps done are relevant to the objective of this project which is to enhance images of ANA in order to extract features from it.

After segmentation is done, feature extraction process is done by using Grayco function in MATLAB. The result for homogeneity, contrast, energy and entropy of the output images are obtained where the highest difference in value of the features will be used to differentiate Nucleolar and Centromere pattern.

By having a GUI in the end of this project ease the usage of this coding at which there will be a simple procedure on how to use this program as compared to the probability at which the operator will mistakenly change the coding in the MATLAB editor and causing the program to not work in the future. GUI is also user-friendly at which the users do not required to attend any MATLAB programming class and can easily be used by anyone.

The objective of this project is achieved which is to do features extraction of the Centromere and Nucleolar pattern with addition of classifying the pattern from one another. Designing the algorithm that will automatically classify the patterns is also done with addition of GUI that will ease the usage of this program.

5.2 Recommendation and Future Work

There is some problem in having extra HEp-2 cell drops compared to the ratio of determining the concentration of the ANA in the blood sample will cause noise and extra filtering required. If there are extra HEp-2 cell drops either on the microscope slide or in the blood sample, the original image will be black and orange instead of having black and green fluorescence. If this situation occurs, the algorithm might process the image differently and the output image obtained might be unclear for further analysis.

The medical technician dealing with the blood sample should be careful in dropping HEp-2 cell during the testing to ensure that high quality image of cells can be obtained and can be used in the algorithm.

In the future, instead of classifying two patterns, more ANA patterns should be included. This is because, there are other ANA patterns existing in human blood sample such as Homogeneus and Speckle which will lead to other diseases. However, if more patterns are included, SVM cannot be used to classify the images as SVM can only classify two groups of images. Research is required to analyze types of other classifier can be used to differentiate the images.

REFERENCES

[1] D. H. HUSSEIN, "Sign of the wolf," in *The Start*, ed. Malaysia: Star Publication (M) Bhd, 2008.

[2] T. Nakabayashi, T. Kumagai, K. Yamauchi, M. Sugano, A. Kuramoto, K. Fujita, *et al.*, "Evaluation of the automatic fluorescent image analyzer, Image Titer, for quantitative analysis of antinuclear antibodies," *Am J Clin Pathol*, vol. 115, pp. 424-9, Mar 2001.

[3] P. Ghosh, S. Dwivedi, S. Naik, V. Agarwal, A. Verma, A. Aggarwal, *et al.*, "Antinuclear antibodies by indirect immunofluorescence : optimum screening dilution for diagnosis of systemic lupus erythematosus," *Indian J Med Res*, vol. 126, pp. 34-8, Jul 2007.

[4] X. Tian, X. Wang, L. Wang, and H. Dong, "Application of digital image processing in the measurement of casting surface roughness," in *Computer Science and Information Processing (CSIP), 2012 International Conference on*, 2012, pp. 19-21.

[5] A. Chaudhary and S. S. Singh, "Lung Cancer Detection on CT Images by Using Image Processing," in *Computing Sciences (ICCS), 2012 International Conference on*, 2012, pp. 142-146.

[6] T. Jiya and P. Yanqin, "Research of the Matlab application in the fingerprint identification system," in *Image Analysis and Signal Processing (IASP), 2012 International Conference on*, 2012, pp. 1-5.

[7] B. a. K. R. C. Ramamurthy, 2012. , "Content based medical image retrieval with texture content using gray level co-occurrence matrix and k-means clustering algorithms. ," *Journal of Computer Science 8 (7)*, vol. 8, pp. 1070-1076, 2012.

[8] J. R. Smith and C. Shih-Fu, "Automated binary texture feature sets for image retrieval," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, 1996, pp. 2239-2242 vol. 4.

[9] L. Bin and P. Fen, "Rotation-invariant texture features extraction using Dual-Tree Complex Wavelet Transform," in *Information Networking and Automation (ICINA), 2010 International Conference on*, 2010, pp. V1-361-V1-364.

[10] M. Saad, "Low Level Color and Texture Feature Extraction for Content Based Image Retrieval," *EE318K: Multi-Dimensional Digital Signal Processing*, May 09, 2008 2008.

[11] P. Babaghorbani, S. Parvaneh, A. R. Ghassemi, and K. Manshai, "Sonography Images for Breast Cancer Texture Classification in Diagnosis of Malignant or Benign Tumors," in *Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference on*, 2010, pp. 1-4.

[12] G.-A. Julio-Cesar, "Region of Interest Extraction Method Using Wavelets," 2009, pp. 119-124.

[13] F. Xu, C. Wang, and H. Zhang, "Region feature extraction based on improved regularization method in SAR image," in *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, 2007, pp. 571-573.

[14] C. Chang Wen, L. Weiwen, Y. Fang-Fang, and C. Lulin, "Portal image feature extraction by hierarchical region processing technique," in *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, 1995, pp. 3561-3566 vol.4.

[15] B. Demir and S. Erturk, "Improving SVM classification accuracy using a hierarchical approach for hyperspectral images," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 2009, pp. 2849-2852.

[16] J. Cervantes, L. XiaoOu, and Y. Wen, "SVM Classification for Large Data Sets by Considering Models of Classes Distribution," in *Artificial Intelligence - Special Session, 2007. MICAI 2007. Sixth Mexican International Conference on*, 2007, pp. 51-60.

[17] A. Mathur and G. M. Foody, "Multiclass and Binary SVM Classification: Implications for Training and Classification Users," *Geoscience and Remote Sensing Letters, IEEE*, vol. 5, pp. 241-245, 2008.

[18] Z. Guoqing and X. Shengyun, "Comparison of object-oriented and Maximum Likelihood Classification of land use in Karst area," in *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, 2012, pp. 6099-6102.

[19] A. Ahmad and S. Quegan, "Analysis of Maximum Likelihood classification technique on Landsat 5 TM satellite data of tropical land covers," in *Control*

System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on, 2012, pp. 280-285.

[20] L. Xin and I. S. Yetik, "A Maximum Likelihood Classification method for image segmentation considering subject variability," in *Image Analysis & Interpretation (SSIAI), 2010 IEEE Southwest Symposium on*, 2010, pp. 125-128.

[21] Z. Shaohua and W. Zhaohua, "Study on solder joint image segmentation techniques based on Matlab," in *Electronic Packaging Technology and High Density Packaging (ICEPT-HDP), 2011 12th International Conference on*, 2011, pp. 1-3.

APPENDICES

APPENDIX A

MATLAB Coding for Color Conversion

```
%% Basic Image Processing
% Declare a figure to display images
figure(1);
% Read image named Centromere with JPEG extension and put it in a variable
named 'Image'
Image = imread('Centromere.jpg');
% Divide the figure into two rows, one colums and choose the first quardent
subplot(2,1,1);
% display the image and give it a title
imshow(Image), title('Original image');

% Convert the image to grayscale and put it in a variable named 'ImageGray'
ImageGray = rgb2gray(Image);
% Display gray image in the second row, first image
subplot(2,3,4);
imshow(ImageGray, []), title('Gray image');
% Compute Image Negative
NegativeImage = 255-ImageGray;
% Display negative image in the second row second image
subplot(2,3,5);
imshow(NegativeImage, []), title('Negative image');

% Compute Log Image
LogImage = log(1+double(ImageGray));
% display negative image in the second row, third image
subplot(2,3,6);
imshow(LogImage, []), title('Log image');
```

APPENDIX B

MATLAB Coding for Histogram Processing

```
%% Histogram Processing
% Display grayscale image
subplot (1,4,1);
Image = imread('Centromere.jpg');
imshow(Image), title ('Original Image');
subplot(2,4,2);
ImageGray = rgb2gray(Image);
imshow(ImageGray, []), title('Gray Image');

% display histogram of the gray image
subplot(2,4,6);
imhist(ImageGray), title('Gray image histogram');

% Image contrast stretching or histogram stretching
StretchedImage = imadjust(ImageGray, stretchlim(ImageGray),
[]);
subplot(2,4,3);
imshow(StretchedImage, []), title('Stretched image');

% display histogram
subplot(2,4,7);
imhist(StretchedImage), title('Stretched image histogram');

% histogram equalization
EqualizedImage = histeq(ImageGray);
subplot(2,4,4);
imshow(EqualizedImage, []), title('Histogram equalized image');

% display equalized histogram
subplot(2,4,8);
imhist(EqualizedImage), title('Equalized image histogram');
```

APPENDIX C

MATLAB Coding for Noise Reduction

```
% Display grayscale image
Image = imread('Centromere.jpg');
ImageGray = rgb2gray(Image);
subplot(3,1,1);
imshow(ImageGray, []), title('Grayscale Image');

% add Gaussian noise of zero mean and std=0.01 to the image
ImageGaussNoise = imnoise(ImageGray, 'gaussian', 0, 0.01);
% display image with Gaussian noise
subplot(3,2,3);
imshow(ImageGaussNoise, []), title('Image with Gaussian Noise');

% define a spatial filter
H = fspecial('gaussian', [5 5], 0.1);
% apply the filter on image
FilteredImage = imfilter(ImageGaussNoise, H);
% Display filtered image
subplot(3,2,5);
imshow(FilteredImage, []), title('Image Filtered with Gaussian Filter');

% add salt&pepper noise of 5% density to the image
ImageGaussNoise = imnoise(ImageGray, 'Salt & Pepper', 0.05);
% display image with Salt&pepper noise
subplot(3,2,4);
imshow(ImageGaussNoise, []), title('Image with Salt & Pepper Noise');

% appling median filter
MedianImage = medfilt2(ImageGray, [5 5]);
% Display median filtered image
subplot(3,2,6);
imshow(MedianImage, []), title('Median Filtered Image');
```

APPENDIX D

MATLAB Coding for Image Processing Process

```
%% Basic Image Processing
% Declare a figure to display images
figure(1);
% Read and image named Centromere with JPEG extension and put it in a
variable
% named 'Image'
Image1 = imread('Centromere.jpg');
% display the image and give it a title
subplot (2,1,1);
imshow(Image1), title('Original image');

%Cropping Image
Image2 = imcrop(Image1);
subplot (2,3,4);
imshow (Image2), title ('Cropped Image');

%Resize Image
Image3 = imresize (Image2, [256 256]);
subplot (2,3,5);
imshow (Image3); title ('Resized Image');

%Sharpen Image
Image4 = imsharpen(Image3, 'Radius',1, 'Amount',2);
subplot (2,3,6);
imshow (Image4); title ('Sharpened Image');
```

APPENDIX E

MATLAB Coding for Entropy Texture Filter

1. RGB Input Image

```
%Read Image
I = imread ('Centromere.jpg');
figure;
subplot(2,1,1);
imshow (I), title ('Original Image');

%Use Entropyfilt Texture Filter
E = entropyfilt (I);
%Rescale Texture Image
Eim = mat2gray (E);
subplot(2,2,3);
imshow (Eim), title ('Entropy Filtered Image');

%Threshold image with 0.8 value
BW1 = im2bw (Eim, 0.8);
subplot (2,2,4);
imshow (BW1), title ('Threshold Texture');
```

2. Grayscale Input Image

```
%Read Image
I = imread ('Centromere.jpg');
figure;
subplot(2,2,1);
imshow (I), title ('Original Image');

ImageGray = rgb2gray (I);
subplot (2,2,2);
imshow (ImageGray), title ('Grayscale Image');

%Use Entropyfilt Texture Filter
E = entropyfilt (ImageGray);
%Rescale Texture Image
Eim = mat2gray (E);
subplot(2,2,3);
imshow (Eim), title ('Entropy Filtered Image');

%Threshold image with 0.8 value
BW1 = im2bw (Eim, 0.8);
subplot (2,2,4);
imshow (BW1), title ('Threshold Texture');
```

APPENDIX F

MATLAB Coding for Standard Filter

1. RGB Input Image

```
%Read Image
I = imread ('Centromere.jpg');
figure;
subplot(1,2,1);
imshow (I), title ('Original Image');

%Use Stdfilt Texture Filter
S = stdfilt (I);
subplot (1,2,2);
imshow (mat2gray(S)), title ('Std Filtered Image');
```

2. Grayscale Input Image

```
%Read Image
I = imread ('Centromere.jpg');
figure;
subplot(2,2,1);
imshow (I), title ('Original Image');

ImageGray = rgb2gray (I);
subplot (2,2,2);
imshow (ImageGray), title ('Grayscale Image');

%Use Stdfilt Texture Filter
S = stdfilt (ImageGray);
subplot (2,1,2);
imshow (mat2gray(S)), title ('Std Filtered Image');
```

APPENDIX G

MATLAB Coding for Range Filtering

1. RGB Input Image

```
%Read Image
I = imread ('Centromere.jpg');
figure;
subplot(1,2,1);
imshow (I), title ('Original Image');

%Use Rangefilt Texture Filter
R = rangefilt (I, ones(5));
subplot (1,2,2);
imshow (R), title ('Range Filtered Image');
```

2. Grayscale Input Image

```
%Read Image
I = imread ('Centromere.jpg');
figure;
subplot(2,2,1);
imshow (I), title ('Original Image');

%Converting Image into Grayscale
ImageGray = rgb2gray (I);
subplot (2,2,2);
imshow (ImageGray), title ('Grayscale Image');

%Use Rangefilt Texture Filter
R = rangefilt (ImageGray, ones(5));
subplot (2,1,2);
imshow (R), title ('Range Filtered Image');
```

APPENDIX H

MATLAB Coding for Gradient Magnitude

```
%% Image Segmentation by Gradient Magnitude
hy = fspecial('sobel');
hx = hy';
Image9 = imfilter(double(Image8), hy, 'replicate');
Image10 = imfilter(double(Image8), hx, 'replicate');
Image11 = sqrt(Image9.^2 + Image10.^2);
Figure (1);
imshow(Image11,[]);
```

APPENDIX I

MATLAB Coding for Cell Cropping

```
%% Basic Image Processing
% Declare a figure to display images
figure(1);
% Read and image named Centromere with JPEG extension and put it in a
variable
% named 'Image'
Image1 = imread('Centromere.jpg');
% display the image and give it a title
subplot (1,2,1);
imshow(Image1), title('Original image');

%Cropping Image
Image2 = imcrop(Image1);
subplot (1,2,2);
imshow (Image2), title ('Cropped Image');
```

APPENDIX J

MATLAB Coding for GLCM

```
%% Texture Feature Extraction using GLCM
GLCM =
graycomatrix(Image7,'NumLevels',9,'G',[]);
Properties = graycoprops (GLCM, 'Energy')
```

APPENDIX K

MATLAB Coding for SVM Classification

```
%% Pattern Classification
% Training Image
E1 = 0.4993;
E2 = 0.4875;
E3 = 0.4874;
E4 = 0.4769;
E5 = 0.4604;
E6 = 0.4577;
E7 = 0.4435;
E8 = 0.4414;
E9 = 0.4404;
E10 = 0.4336;
E11 = 0.4219;
E12 = 0.4185;
E13 = 0.4184;
E14 = 0.4123;
E15 = 0.4051;
E16 = 0.4038;
E17 = 0.3999;
E18 = 0.3943;
E19 = 0.3903;
E20 = 0.3875;
E21 = 0.3863;
E22 = 0.3807;
E23 = 0.3747;
E24 = 0.3705;
E25 = 0.3673;
E26 = 0.3659;
E27 = 0.3659;
E28 = 0.3620;
E29 = 0.3591;
E30 = 0.3561;
E31 = 0.3518;
E32 = 0.3512;
E33 = 0.3502;
E34 = 0.3492;
E35 = 0.3485;
E36 = 0.3461;
E37 = 0.3427;
E38 = 0.3412;
E39 = 0.3404;
E40 = 0.3370;
E41 = 0.3363;
E42 = 0.3298;
E43 = 0.3265;
E44 = 0.3243;
E45 = 0.3233;
E46 = 0.3225;
E47 = 0.3194;
```

```
E48 = 0.3139;
E49 = 0.3128;
E50 = 0.3127;
E51 = 0.3094;
E52 = 0.3088;
E53 = 0.3074;
E54 = 0.3035;
E55 = 0.3027;
E56 = 0.3008;
E57 = 0.2995;
E58 = 0.2985;
E59 = 0.2957;
E60 = 0.2951;
E61 = 0.2933;
E62 = 0.2821;
E63 = 0.2814;
E64 = 0.2797;
E65 = 0.2565;
E66 = 0.2489;
E67 = 0.2453;
E68 = 0.2401;
E69 = 0.2337;
E70 = 0.2148;
E71 = 0.2012;
E72 = 0.2004;
E73 = 0.1984;
E74 = 0.1750;
E75 = 0.1707;
E76 = 0.1511;
E77 = 0.5243;
E78 = 0.5335;
E79 = 0.5342;
E80 = 0.5512;
E81 = 0.5602;
E82 = 0.5643;
E83 = 0.5702;
E84 = 0.5826;
E85 = 0.6304;
E86 = 0.6327;
E87 = 0.6474;
E88 = 0.6583;
E89 = 0.6643;
E90 = 0.6646;
E91 = 0.6734;
E92 = 0.6785;
E93 = 0.6925;
E94 = 0.7021;
E95 = 0.7045;
E96 = 0.7166;
E97 = 0.7185;
E98 = 0.7235;
E99 = 0.7291;
E100 = 0.7311;
E101 = 0.7360;
```

```
E102 = 0.7580;
E103 = 0.7602;
E104 = 0.7626;
E105 = 0.7701;
E106 = 0.7704;
E107 = 0.7712;
E108 = 0.7802;
E109 = 0.7903;
E110 = 0.7927;
E111 = 0.8112;
E112 = 0.8168;
E113 = 0.8332;
E114 = 0.8334;
E115 = 0.8435;
E116 = 0.8526;
E117 = 0.8550;
E118 = 0.8938;
Training =
cat(1,E1,E2,E3,E4,E5,E6,E7,E8,E9,E10,E11,E12,E13,E14,E15,E16,E17,E18,E19,E2
0,E21,E22,E23,E24,E25,E26,E27,E28,E29,E30,E31,E32,E33,E34,E35,E36,E37,E38,E
39,E40,E41,E42,E43,E44,E45,E46,E47,E48,E49,E50,E51,E52,E53,E54,E55,E56,E57,
E58,E59,E60,E61,E62,E63,E64,E65,E66,E67,E68,E69,E70,E71,E72,E73,E74,E75,E76
,E77,E78,E79,E80,E81,E82,E83,E84,E85,E86,E87,E88,E89,E90,E91,E92,E93,E94,E9
5,E96,E97,E98,E99,E100,E101,E102,E103,E104,E105,E106,E107,E108,E109,E110,E1
11,E112,E113,E114,E115,E116,E117,E118);

% Define Pattern for Each Energy: 0 = Centromere & 1 = Nucleolar
Group =
[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
;0;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;
1;1;1;1;1;1];
ImageTrain = svmtrain(Training,Group);

% Classifying Test Image
Pattern = svmclassify(ImageTrain,Energy);
if Pattern == 0
  disp('Centromere')
else
  disp('Nucleolar')
end
```

APPENDIX L

Results from GLCM Features Extraction

Energy		Homogeneity		Contrast		Correlation	
Centromere	**Nucleolar**	**Centromere**	**Nucleolar**	**Centromere**	**Nucleolar**	**Centromere**	**Nucleolar**
0.4993	0.5243	0.9438	0.9140	0.9172	0.0570	0.9633	0.8059
0.4875	0.5335	0.9424	0.9158	0.8829	0.0654	0.9631	0.8245
0.4874	0.5342	0.9394	0.9168	0.4462	0.0756	0.9609	0.8250
0.4769	0.5512	0.9372	0.9181	0.3783	0.0800	0.9566	0.8254
0.4604	0.5602	0.9322	0.9182	0.3751	0.0850	0.9535	0.8488
0.4577	0.5643	0.9310	0.9224	0.3705	0.0889	0.9524	0.8591
0.4435	0.5702	0.9283	0.9283	0.3666	0.0919	0.9488	0.8593
0.4414	0.5826	0.9280	0.9304	0.3438	0.0931	0.9476	0.8653
0.4404	0.6304	0.9272	0.9311	0.3425	0.1037	0.9464	0.8690
0.4336	0.6327	0.9249	0.9396	0.3397	0.1048	0.9452	0.8724
0.4219	0.6474	0.9242	0.9402	0.3299	0.1051	0.9448	0.8731
0.4185	0.6583	0.9235	0.9405	0.3187	0.1058	0.9446	0.8753
0.4184	0.6643	0.9213	0.9427	0.3181	0.1059	0.9435	0.8835
0.4123	0.6646	0.9195	0.9437	0.3148	0.1095	0.9406	0.8840
0.4051	0.6734	0.9191	0.9453	0.3144	0.1122	0.9401	0.8842
0.4038	0.6785	0.9182	0.9463	0.3090	0.1153	0.9400	0.8844
0.3999	0.6925	0.9182	0.9471	0.3022	0.1181	0.9393	0.8867
0.3943	0.7021	0.9176	0.9514	0.2957	0.1193	0.9382	0.8877
0.3903	0.7045	0.9176	0.9517	0.2926	0.1298	0.9374	0.8880
0.3875	0.7166	0.9172	0.9520	0.2888	0.1305	0.9369	0.8937
0.3863	0.7185	0.9158	0.9535	0.2864	0.1316	0.9354	0.8948
0.3807	0.7235	0.9145	0.9556	0.2769	0.1319	0.9351	0.8948
0.3747	0.7291	0.9143	0.9557	0.2750	0.1330	0.9349	0.8952
0.3705	0.7311	0.9142	0.9559	0.2735	0.1366	0.9348	0.8977
0.3673	0.7360	0.9126	0.9569	0.2717	0.1399	0.9348	0.8989
0.3659	0.7580	0.9121	0.9571	0.2680	0.1460	0.9326	0.8989
0.3659	0.7602	0.9115	0.9581	0.2674	0.1484	0.9324	0.9012
0.3620	0.7626	0.9115	0.9588	0.2609	0.1501	0.9320	0.9014
0.3591	0.7701	0.9109	0.9595	0.2593	0.1552	0.9314	0.9027
0.3561	0.7704	0.9108	0.9605	0.2560	0.1563	0.9314	0.9027
0.3518	0.7712	0.9101	0.9609	0.2547	0.1659	0.9304	0.9076
0.3512	0.7802	0.9098	0.9620	0.2533	0.1740	0.9303	0.9079
0.3502	0.7903	0.9097	0.9639	0.2526	0.1755	0.9303	0.9080
0.3492	0.7927	0.9075	0.9653	0.2453	0.1773	0.9298	0.9082
0.3485	0.8112	0.9069	0.9690	0.2449	0.1789	0.9291	0.9091
0.3461	0.8168	0.9056	0.9696	0.2446	0.2096	0.9290	0.9092
0.3427	0.8332	0.9030	0.9722	0.2402	0.2129	0.9285	0.9118
0.3412	0.8334	0.9018	0.9726	0.2328	0.2150	0.9278	0.9130
0.3404	0.8435	0.9013	0.9734	0.2300	0.2217	0.9274	0.9171
0.3370	0.8526	0.9009	0.9748	0.2259	0.2217	0.9271	0.9216
0.3363	0.8550	0.8998	0.9766	0.2237	0.2296	0.9258	0.9317
0.3298	0.8938	0.8973	0.9829	0.2208	0.2400	0.9247	0.9334
0.3265		0.8952		0.2163		0.9230	
0.3243		0.8952		0.2142		0.9227	
0.3233		0.8951		0.2108		0.9220	
0.3225		0.8938		0.2099		0.9212	
0.3194		0.8935		0.2047		0.9205	
0.3139		0.8934		0.2015		0.9187	
0.3128		0.8927		0.2011		0.9183	
0.3127		0.8925		0.2004		0.9176	
0.3094		0.8921		0.1996		0.9169	
0.3088		0.8916		0.1989		0.9151	
0.3074		0.8904		0.1988		0.9109	
0.3035		0.8900		0.1987		0.9101	
0.3027		0.8890		0.1952		0.9001	
0.3008		0.8881		0.1943		0.8993	

0.2995		0.8881		0.1899		0.8948	
0.2985		0.8877		0.1897		0.8901	
0.2957		0.8836		0.1892		0.8840	
0.2951		0.8829		0.1867		0.8825	
0.2933		0.8818		0.1862		0.8824	
0.2821		0.8818		0.1810		0.8798	
0.2814		0.8799		0.1799		0.8731	
0.2797		0.8771		0.1792		0.8718	
0.2565		0.8736		0.1777		0.8704	
0.2489		0.8699		0.1762		0.8695	
0.2453		0.8681		0.1698		0.8677	
0.2401		0.8660		0.1680		0.8654	
0.2337		0.8617		0.1671		0.8632	
0.2148		0.8609		0.1614		0.8609	
0.2012		0.8591		0.1570		0.8557	
0.2004		0.8568		0.1516		0.8498	
0.1984		0.8292		0.1498		0.8482	
0.1750		0.3705		0.1387		0.8480	
0.1707		0.3561		0.1335		0.8407	
0.1511		0.3461		0.1239		0.8388	

APPENDIX M

Gantt Chart

Project Activities	Weeks																												
	FYP I														FYP II														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Project Title Confirmation																													
Initial Research																													
MATLAB Image Processing Class																													
Project Understanding																													
Submission of Extended Proposal																													
Details Research																													
Technique Identification and Study																													
Proposal Defense Presentation																													
Image Pre-processing																													
Image Segmentation																													
Interim Report Submission																													
Image Features Extraction																													
Result Analysis and Discussion part 1																													
Algorithm Design																													
Result Analysis and Discussion part 2																													
Submission of Progress Report																													
Preparation of Pre-SEDEX																													
Pre-SEDEX																													
Submission of Draft Report																													
Submission of Technical Report																													
Submission of Final Report																													
VIVA																													

APPENDIX N

MATLAB Final Coding (Algorithm)

```
%% Basic Image Pre-Processing
% named 'Image'
disp ('Please Enter Image Name Followed by Type of File. Example : Centromere.jpg')
prompt = 'Image Name: ';
Image = input (prompt, 's');

% Read and image named Centromere with JPEG extension and put it in a variable
Image1 = imread(Image);
figure(1);
subplot (2,2,1);
imshow(Image1), title('Original image');

% Convert RGB Image to Grayscale Image
Image2 = rgb2gray(Image1);
subplot(2,2,2);
imshow(Image2, []), title('Grayscale image');

% Compute Stretch Image Histogram
Image3 = imadjust(Image2, stretchlim(Image2), []);
subplot(2,3,4);
imshow(Image3, []), title('Stretched Image');

% Add Wiener Filter to the Image
Image4 = wiener2(Image3,[20 20]);
subplot(2,3,5);
imshow (Image4), title('Wiener Filtered Image');

% Apply Gaussian Filter on image
h = fspecial('gaussian', [3 3], 0.5);
subplot (2,3,6);
Image5 = imfilter(Image4, h);
imshow(Image5, []), title('Gaussian Image');

Image6 = imadjust (Image5);
figure (2);
imshow (Image6), title ('Contrast Adjusted Image');

%Cropping Image
Image7 = imcrop(Image6);
figure (3);
subplot (1,2,1);
imshow (Image7), title ('Cropped Image');

Image8 = imadjust (Image7);
subplot (1,2,2);
imshow (Image8), title ('Contrast Adjusted Image');
```

```
%% Image Segmentation by Gradient Magnitude

hy = fspecial('sobel');
hx = hy';
Image9 = imfilter(double(Image8), hy, 'replicate');
Image10 = imfilter(double(Image8), hx, 'replicate');
Image11 = sqrt(Image9.^2 + Image10.^2);
figure(4);
imshow(Image11,[]), title('Gradient Magnitude')

%% Texture Feature Extraction using GLCM
GLCM = graycomatrix(Image11,'NumLevels',9,'G',[]);
Properties = graycoprops (GLCM)

%% Pattern Classification
% Training Image
E1 = 0.4993;
E2 = 0.4875;
E3 = 0.4874;
E4 = 0.4769;
E5 = 0.4604;
E6 = 0.4577;
E7 = 0.4435;
E8 = 0.4414;
E9 = 0.4404;
E10 = 0.4336;
E11 = 0.4219;
E12 = 0.4185;
E13 = 0.4184;
E14 = 0.4123;
E15 = 0.4051;
E16 = 0.4038;
E17 = 0.3999;
E18 = 0.3943;
E19 = 0.3903;
E20 = 0.3875;
E21 = 0.3863;
E22 = 0.3807;
E23 = 0.3747;
E24 = 0.3705;
E25 = 0.3673;
E26 = 0.3659;
E27 = 0.3659;
E28 = 0.3620;
E29 = 0.3591;
E30 = 0.3561;
E31 = 0.3518;
E32 = 0.3512;
E33 = 0.3502;
E34 = 0.3492;
E35 = 0.3485;
E36 = 0.3461;
E37 = 0.3427;
E38 = 0.3412;
```

```
E39 = 0.3404;
E40 = 0.3370;
E41 = 0.3363;
E42 = 0.3298;
E43 = 0.3265;
E44 = 0.3243;
E45 = 0.3233;
E46 = 0.3225;
E47 = 0.3194;
E48 = 0.3139;
E49 = 0.3128;
E50 = 0.3127;
E51 = 0.3094;
E52 = 0.3088;
E53 = 0.3074;
E54 = 0.3035;
E55 = 0.3027;
E56 = 0.3008;
E57 = 0.2995;
E58 = 0.2985;
E59 = 0.2957;
E60 = 0.2951;
E61 = 0.2933;
E62 = 0.2821;
E63 = 0.2814;
E64 = 0.2797;
E65 = 0.2565;
E66 = 0.2489;
E67 = 0.2453;
E68 = 0.2401;
E69 = 0.2337;
E70 = 0.2148;
E71 = 0.2012;
E72 = 0.2004;
E73 = 0.1984;
E74 = 0.1750;
E75 = 0.1707;
E76 = 0.1511;
E77 = 0.5243;
E78 = 0.5335;
E79 = 0.5342;
E80 = 0.5512;
E81 = 0.5602;
E82 = 0.5643;
E83 = 0.5702;
E84 = 0.5826;
E85 = 0.6304;
E86 = 0.6327;
E87 = 0.6474;
E88 = 0.6583;
E89 = 0.6643;
E90 = 0.6646;
E91 = 0.6734;
E92 = 0.6785;
```

```
E93 = 0.6925;
E94 = 0.7021;
E95 = 0.7045;
E96 = 0.7166;
E97 = 0.7185;
E98 = 0.7235;
E99 = 0.7291;
E100 = 0.7311;
E101 = 0.7360;
E102 = 0.7580;
E103 = 0.7602;
E104 = 0.7626;
E105 = 0.7701;
E106 = 0.7704;
E107 = 0.7712;
E108 = 0.7802;
E109 = 0.7903;
E110 = 0.7927;
E111 = 0.8112;
E112 = 0.8168;
E113 = 0.8332;
E114 = 0.8334;
E115 = 0.8435;
E116 = 0.8526;
E117 = 0.8550;
E118 = 0.8938;
Training                                                                =
cat(1,E1,E2,E3,E4,E5,E6,E7,E8,E9,E10,E11,E12,E13,E14,E15,E16,E17,E18,E19,E2
0,E21,E22,E23,E24,E25,E26,E27,E28,E29,E30,E31,E32,E33,E34,E35,E36,E37,E38,E
39,E40,E41,E42,E43,E44,E45,E46,E47,E48,E49,E50,E51,E52,E53,E54,E55,E56,E57,
E58,E59,E60,E61,E62,E63,E64,E65,E66,E67,E68,E69,E70,E71,E72,E73,E74,E75,E76
,E77,E78,E79,E80,E81,E82,E83,E84,E85,E86,E87,E88,E89,E90,E91,E92,E93,E94,E9
5,E96,E97,E98,E99,E100,E101,E102,E103,E104,E105,E106,E107,E108,E109,E110,E1
11,E112,E113,E114,E115,E116,E117,E118);

% Define Pattern for Each Energy: 0 = Centromere & 1 = Nucleolar
Group                                                                   =
[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
;0;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;
1;1;1;1;1;1];
ImageTrain = svmtrain(Training,Group);

% Classifying Test Image
Pattern = svmclassify(ImageTrain,Energy);
if Pattern == 0
  disp('Centromere')
else
  disp('Nucleolar')
end
```