DETECTION AND SUMMATION OF PUS CELL FOR SPUTUM QUALITY TESTING

NORAZURA BINTI ABDUL HALIM

A thesis submitted in fulfillment

of requirements for the award of the degree of

Bachelor of Electrical Engineering (Electronics)

Faculty of Electrical and Electronics Engineering

Universiti Malaysia Pahang

JUNE 2012

SUPERVISOR DECLARATION

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature : _____

Name : MADAM ROSYATI BINTI HAMID

Date : 21 JUNE 2012

STUDENT DECLARATION

I declare that this thesis entitled " *Detection and Summation of Pus Cell for Sputum Quality Testing* " is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : .....

Name : NORAZURA BINTI ABDUL HALIM

Date : 21 JUNE 2012

ACKNOWLEDGEMENT

I would like to thanks Allah s.w.t. for giving me strength to complete this project. Without His bless, this project will never be completed as what have been planned.

Besides, I would like to thanks my supervisor, Mdm Rosyati Hamid, family and friend for helping, guiding and supporting me a lot to finish up this project. This project took almost one year to be finished. So, without support from them, this project will be impossible to be completed on time.

Since this project has a collaboration with Universiti Sains Malaysia Kubang Kerian, I would like to thank Prof. Madya Dr. Zeehaida Mohamed for providing me with a lot of information regarding pus cells and how to determine the quality of sputum based on Modified Bartlett's Criteria. Without this information, it will be hard for me to understand this project since it relates to medical field.

Finally, I would like to thank once again to my supervisor for trusting me to handle this project and personally, this project really create a lot of beautiful memories to me.

ABSTRAK

Penyakit yang berkaitan dengan paru-paru seperti *Moraxella catarrhalis*, *Mycobacterium tuberculosis* dan lain-lain boleh diketahui melalui kahak. Walaubagaimanapun, sampel kahak perlu melalui kultur proses yang menelan belanja yang tinggi sebelum penyakit-penyakit di atas dapat diketahui. Maka, ujian terhadap kualiti kahak harus dijalankan untuk mengelakkan berlakunya sebarang pembaziran. Hanya kahak yang berkualiti atau positif sahaja yang akan menjalani proses ini. Projek ini dijalankan untuk menggantikan kaedah manual yang diamalkan di Hospital Universiti Sains Malaysia (HUSM) untuk menentukan kualiti kahak berdasarkan 'Bartlett Criteria'. Kaedah manual merujuk kepada penilaian kualiti sesuatu kahak dengan melihat sampel kahak tersebut melalui mikroskop. Jurumakaml akan mengira bilangan sel nanah yang terdapat di dalam sampel kahak melalui mikroskop untuk memenuhi 'Bartlett Criteria'. Maka, satu system berdasarkan pemprosesan imej yang mampu untuk mengesan dan mengira bilangan sel nanah secara automatik dibangunkan untuk menggantikan kaedah manual ini. Sistem ini memerlukan empat imej kahak yang mewakili satu sampel kahak. Kesemua imej ini akan diproses satu per satu melalui sistem ini. Sistem ini juga akan mengira bilangan sel nanah yang terdapat di dalam setiap imej kahak. Akhirnya, sistem ini akan memberikan bilangan purata sel nanah bagi empat imej kahak ini dan menentukan skor bagi sel nanah berdasarkan nilai purata yang diperolehi. Skor ditentukan dengan merujuk kepada 'Bartlett Criteria'.

ABSTRACT

Diseases relate to lung such as *Moraxella catarrhalis*, *Mycobacterium tuberculosis* and others can be determined from sputum. However, sputum sample needs to undergo culturing process which requires high cost before the diseases can be determined. Therefore, sputum quality testing is requires to be performed on sputum sample to avoid any waste. So, only the quality or positive sample is cultured and reject the negative sample. This project is conducted to replace manual method used to determine the quality of sputum in USM, Kubang Kerian based on Modified Bartlett's Criteria. The manual method refers to the process of evaluating sputum sample by observing the sample through microscope. The technologists calculate the number of pus cells and epithelial cells through microscope to find out the score for each type of cells according to Bartlett's Criteria. So, vision system based on image processing which is able to detect and count the number of pus cells automatically is developed to enhance the manual method. This system requires at least four images of sputum from one sputum sample. The images are processed one by one through this system. The number of pus cells for each image is determined. At the end, the average number of pus cells for these four images is determined as well as its score. The score is determined by referring to the Bartlett's Criteria.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

Sputum is a material coughed up from the lungs and expectorated through the mouth. Sputum contains pus cells, squamous ephitelial cells, gram-positive and gram-negative organisms. All of these elements are important to determine the quality of sputum. Sputum needs to undergo quality testing before it can be cultured to avoid high cost of culturing process if it is not a quality sputum. Culturing process is performed to detect diseases relates to lung such as *Moraxella catarrhalis*, *Mycobacterium tuberculosis* and others. Only quality sputum is allowed to undergo this process. In USM Kubang Kerian, detection and summation of these elements is manually done by human. The detection and summation is based on Modified Bartlett's criteria. Each criteria of the elements above has its own score where the total of the score will give the result of sputum quality. However, all of these elements need to be detected and counted separately. Human's eyes cannot observe through microscope for a long period of time. Thus, after 10 to 12 samples, the technologists need to rest their eyes. Therefore, automated detection and summation of pus cell for sputum quality testing system is developed to detect and count these elements automatically. However, this project just covers task to detect and count the number of pus cells in sputum image. The image was taken from x10 magnification digital biological microscope. Therefore, MATLAB R2010a is used to develop this automated detection and summation of pus cell for sputum quality testing system. Through image processing tools, some techniques are applied for detection purpose like k-means clustering, circular average filter, dilation and any other techniques which are suitable to remove

the unwanted elements in sputum image. Unwanted elements refer to squamous epithelial cells, gram-positive and gram-negative organisms. After all of these elements have been removed, the left pus cells is counted automatically.

1.1 Problem Statement

There are a lot of criteria that can be chosen to determine the quality of sputum. In HUSM, they are using Modified Bartlett's Criteria for this purpose. Based on this criteria, the number of pus cells in the image need to be determined to get the right score. Technologist in HUSM determine the number of pus cell by calculating these cells manually from sputum image. Since human might have eye drowsiness while dealing with this task, thus automated detection and summation of pus cell for sputum quality testing system needs to be developed to curb this problem. Besides that, this system can speed up and smooth the process. As a result, more than 100 samples can be processed in a day.

1.2 Objective

The objectives of this project are:

I. To develop system which is able to detect and count the number of pus cells automatically.

II. To determine the score of pus cells based on Bartlett's Criteria.

1.3 Scope of Project

This project just covers software development. Most of the techniques used are based on image processing techniques that can be applied by using MATLAB R2010a. The image taken from x10 magnification digital biological microscope is processed to get the image of pus cells from the sputum image that contains pus cells, epithelial cell, bacteria and other unwanted objects. All of the coding of this process is written on M-File command window whereas the results are shown through the final figure. The number of pus cells is also shown in final figure. However, Graphical User Interface (GUI) is developed to present this project properly. Thus, the focuses of this project are:

I. Detect and count the number of pus cells in sputum image.

II. Determine the score of pus cell in sputum sample based on Modified Bartlett's Criteria.

CHAPTER II

LITERATURE REVIEW

Based on *A.S.Abdul Nasir* writing, he states that leukaemia is a blood cancer that causes more deaths than any others cancer among children and young adults under the age of 20. There are two major types of acute luekaemia such as Acute Lymphobastic Leukaemia (ALL) and Acute Myelogenous Leukaemia (AML). In leukaemia diagnosis, size and shape of abnormal white blood cell (WBC) would be observed by haematologists in order to differentiate the types of acute leukaemia.

For the morphological analysis of acute leukaemia images, *A.S.Abdul Nasir* proposed the combination between linear contrast technique and colour segmentation based on HSI (Hue, Saturation, Intensity). This type of colour space was used in order to obtain a fully segmented abnormal WBC and nucleus of acute leukaemia images. Then, k-means clustering algorithm is used to ease the segmentation process. Next, the fully segmented WBC which consists of cytoplasm and nucleus regions can be achieved by using the combination of linear contrast technique and segmentation based on H component image. Meanwhile, the fully segmented nucleus can be obtained by applying the segmentation based on S component image.

These techniques have produced a better effect on improving the accuracy of WBC segmentation with segmentation accuracies of 99.02% and 99.05% for segmented WBC and nucleus, respectively.

2.1 Sputum Quality Testing

Previous study has shown that the implementation of rejection criteria in clinical specimens saves the reagent cost up to US$28000 and 1082 hours of technologist time [1]. It means that culturing process requires high cost and takes time to be performed. Thus, sputum should undergo sputum quality testing before it can be cultured. Then, only sputum with quality is allowed to be cultured. It is done to reduce high cost of culturing process and save technologist time because the culturing process on non quality sputum will waste the cost and time due to zero output. There are six different criteria for judging the acceptability of sputum specimens [11]. All of the criteria, method and criteria for acceptability are summarized in the table1.

Table 2.1: Summary of six published criteria for judging acceptability of sputum specimens.

Criteria	Method	Criteria for acceptability
Bartlett	Assign + and – values: +2 if >25 Neutrophils;+1 if 10-25 Neutrophils; +1 if mucus seen; -2 if >25 EPI;-1 if 10-25 EPI	Any positive score (sum of + and – values assigned)
Murray and Washington	Average no. of EPI/LPF	<10 EPI/LPF
Geckler et al.	Average no. of EPI/LPF	<25 EPI/LPF
Van Scoy	Average no. of Neutrophils/LPF	>25 Neutrophils/LPF
Barry	Assign + and - values: +3 if >150 Neutrophils; +2 if 76-150 Neutrophils; +1 if 1-75 Neutrophils; -3 if >25 EPI; -2 if 16-25 EPI; -1 if 5-15 EPI	Any positive score (sum of + and – values assigned)
Heineman and Radano	Average ratio of Neutrophils to EPI	>10 Neutrophils/EPI

In which,

Neutrophils= pus cells

EPI=squamous epithelial cell

LPF = Low Power Field

2.2 Modified Bartlett's Criteria

The quality of sputum is determined by using Modified Bartlett's criteria in USM Kubang Kerian. Since this project has collaboration with USM Kubang Kerian, thus Modified Bartlett's Criteria is also applied to develop this automatic vision system. Modified Bartlett's Criteria is chosen due to its easiness in interpretation and lower rejection rate than other criteria, thus minimizing the number of missed potential pathogen [1]. Bartlett's criteria is based on the relative number of inflammatory cells, squamous epithelial cells and mucus seen in Gram-stained smears. However, there are some differences in Modified Bartlett's Criteria, whereby the macroscopic appearances of the sample were taken into account:wether they are mucoid, mucopurelent, purelent or blood stained [1]. This criteria uses total score of each criteria to determine the quality of sputum. The score of each criteria is summarized in the Table 2.2.

Table 2.2: Modified Bartlett's Criteria

	Criteria	Score
Neutrophils (pus cells) count (Score A)	< 10 neutrophil/10x field	0
	10-25 neutrophils/10x field	+1
	>25 neutrophils/10x field	+2
Macroscopy (Score B)	Mucoid, Mucopurelent, Purelent, or Blood stained	+1
Squamous epithelial cell count (Score C)	< 10 Squamous epithelial cell/10x field	0
	10-25 Squamous epithelial cell /10x field	-1
	>25 Squamous epithelial cell /10x field	-2

The decision either sputum sample can be cultured or not is depends on the total score. Total score refers to the summation of score A, score B and score C (Total score = scoreA + scoreB +scoreC). The sputum will be cultured if total score is 1 and above. The sputum is rejected for the total score of 0 and below which can be classified as non quality sputum.

2.3 Pus Cells

Eventhough sputum contains pus cells, squamous epithelial cells and other bacteria but this project just focus on detection and summation of pus cell in sputum image. Thus, most of the explanation will relate to pus cells. Pus cell or Polymorphonuclear Neutrophils (PMNs) play a central role in innate immunity, where they dominate the response to infections, in particular in the cystic fibrosis lung. PMNs are phagocytic cells that produce a wide range of antimicrobial agents aimed at killing invading bacteria [12]. PMNs migrate from the blood stream to the injured tissues where they eliminate invading organisms by phagocytosis and killing [5]. The size of pus cell is

smaller than squamous epithelial cell [6]. The example of pus cell, squamous epithelial cell and bacteria in sputum image are shown in Figure 2.1.

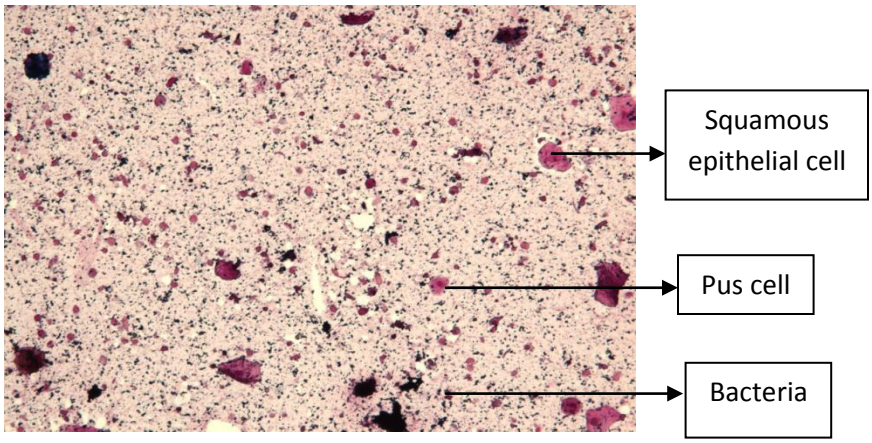


Figure 2.1 Sputum image under x10 computerized microscope.

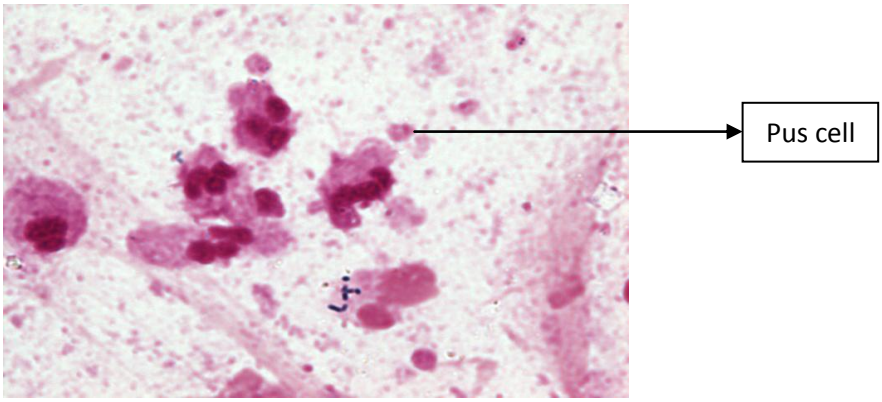
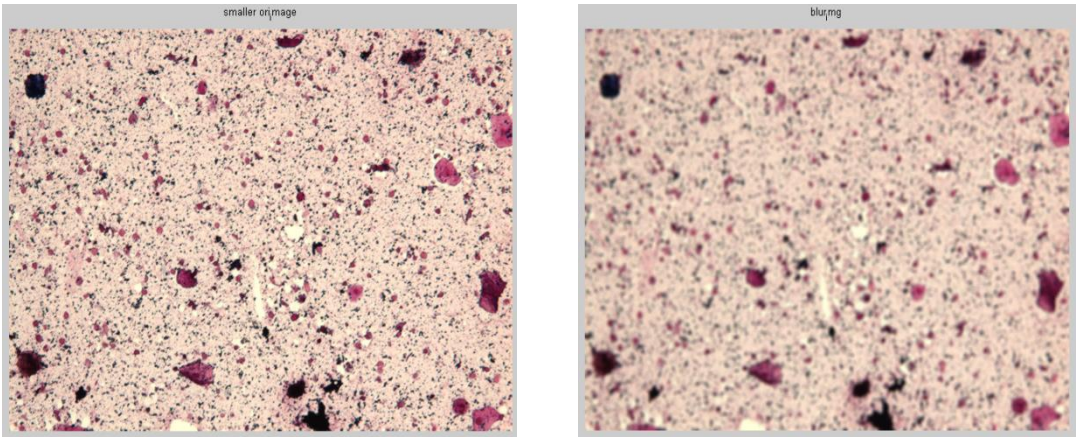


Figure 2.2 Pus cell image under x100 computerized microscope.

2.4 Image Enhancement: Circular Average Filtering

Out of focus blur is usually modeled as circular averaging filter (pillbox) [7]. In MATLAB, this filtering process is done by the command fspecial('disk', radius) that

returns the circular averaging filter (pillbox) within the square matrix of side 2*radius+1. The default radius is 5 [8]. This command softens the edges of the function, alleviating the discrete nature the circular would otherwise demonstrate. Discretely defined circles tend to have edges that are jagged. These jagged edges introduce frequency components that would not be present in a real circular aperture. These frequencies act as noise in the pattern and corrupt the frequency space diagram. To avoid this problem, the 'soft' edged circular should be used. The example of circular averaging filter performed on image is shown in figure 2.



(a) (b)

Figure 2.3 (a) Original sputum image, (b) Image after circular average filtering with radius 3 is performed.

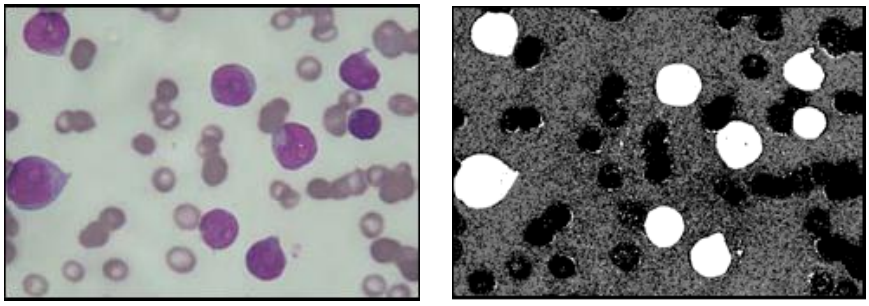
2.5 Colour Segmentation: K-Means Clustering Algorithm

K-Means algorithm classifies the input data points into multiple classes based on their inherent distance from each other. The algorithm assumes that the data features form a vector space and tries to find natural clustering in them [2].

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2 \quad (2.1)$$

Where there are k clusters $s_{i,,}$ $i = 1,2,\ldots,k$ *and* μ_i is the centroid or mean point of all the points $x_j \in s_i$

The cluster should exhibit two properties, they are (1) each group must contain at least one object (2) each object must belong to exactly one group [3]. K-means clustering is used because it is simple and has relatively low computational complexity [4]. In addition, it is suitable for biomedical image segmentation as the number of clusters (K) is usually known for images of particular regions of human anatomy [4]. For smaller values of k the k-Means algorithms give good results [2]. For larger values of k, the segmentation is very coarse, many clusters appear in the images at discrete places [2]. This is because Euclidean distance is not a very good metric for segmentation processes [2]. The example of k-Means clustering is shown in Figure 2.4 with k=3.



(a) (b)

Figure 2.4 (a) original image (b) image after k- Means clustering (three clusters)

2.6 Morphological Operation: Dilation

Dilation 'grows' or 'thickens' objects in a binary image. The specific manner and extent of this thickening is controlled by the shape of the structuring element used [9]. The dilation of A by B, denoted $A \oplus B$ is defined as

$$A \oplus B = \{z|(B)_z \cap A \neq \phi\} \quad (2.2)$$

Where Φ is empty set and B is structure element [10].Dilation operation is shown [10] in figure 2.5.

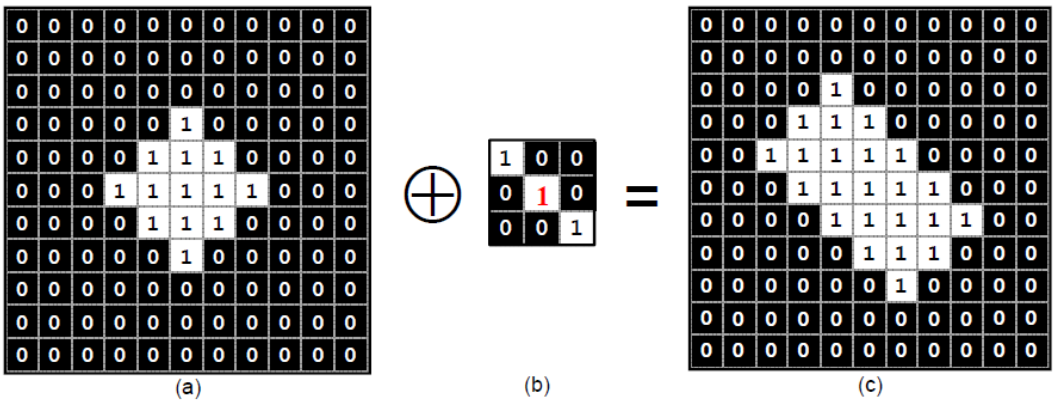


Figure 2.5 Illustration of morphological operation. (a) Original binary image with diamond object, (b) Structure element with three pixels arranged in a diagonal line, the origin of structure is identified by red 1, (c) Dilated image.

Example of dilation operation [10]:

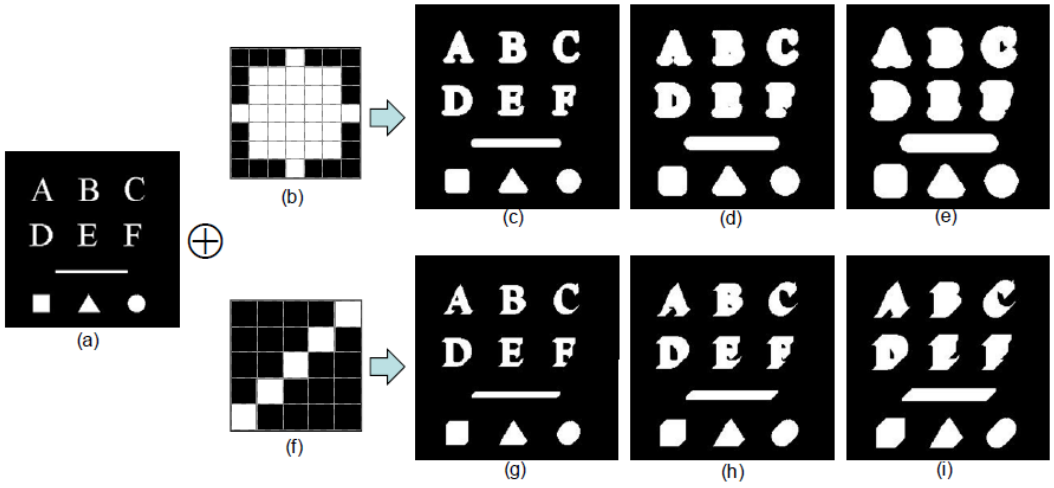


Figure 2.6 Example of morphological operation, (a) binary input image, (b) a disk structure element, (c) Dilated image of (a) by SE (b), (d) after twice dilation of SE (b), (e) after three times dilation of SE (b), (f) a line structure element, (g) dilated image of (a) by SE (f), (h) after twice dilation by SE (f), (i) after three times dilation by SE (f).

2.7 Edge Detection

Edge detection is the approach for segmenting image based on local changes in intensity. It is the most common approach for detecting meaningful discontinuities in gray level [13]. Such continuities are detected by using first and second-order derivatives. The first order derivatives of choice in image processing is the gradient of a 2-D function, *f(x,y),* is defined as the vector [13]

$$\nabla f \equiv grad(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix} \tag{2.3}$$

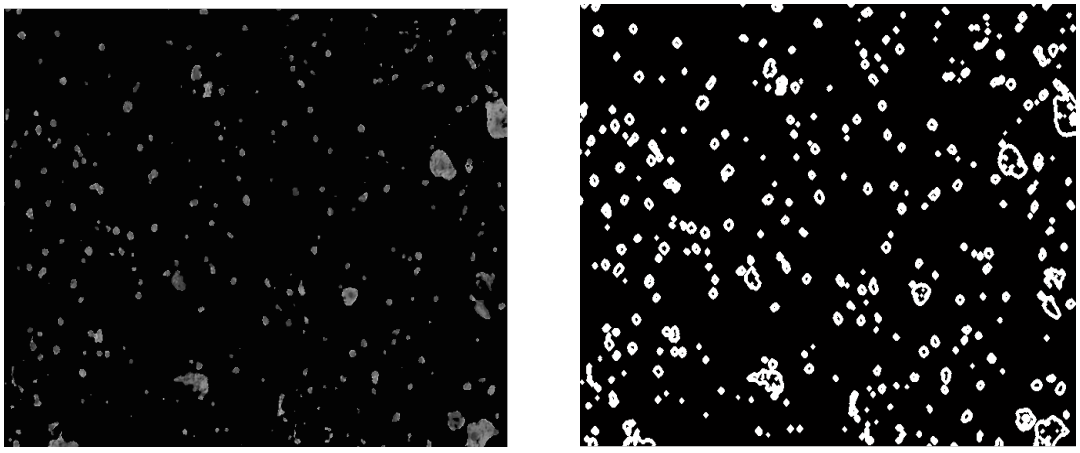
The general syntax for this function is [g,t] = edge(f, 'method', parameters) [13]

In MATLAB R2010a, there are some types of edge detector that can be used to detect edge in image. The summary of these edge detectors is listed in the table 2.3 [13]

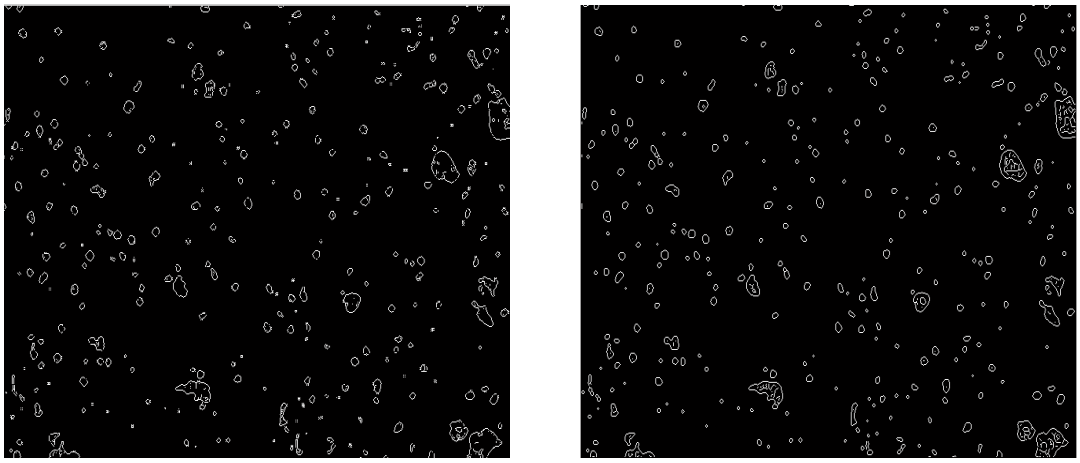
Table 2.3: Edge detection available in function edge.

Edge Detector	Basic Properties
Sobel	Find edges using the Sobel approximation to the derivative.
Prewitt	Find edges using the Prewitt approximation to the derivative.
Roberts	Find edges using the Roberts approximation to the derivative.
Laplacian of a Gaussian (LoG)	Finds edges by looking for zero crossings after filtering *f(x,y)* with a Laplacian of Gaussian filter.
Zero crossings	Finds edges by looking for zero crossings after filtering *f(x,y)* with a user specified filter.
Canny	Finds edges by looking for local maxima of the gradient of I. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be fooled by noise, and more likely to detect true weak edges.

Examples of images after edge detection is performed are shown in figure 2.7



(a) (b)



(c) (d)

Figure 2.7 Examples of edge detection on image. (a) gray image, (b) edge detection ‘sobel’, (c) edge detection ‘prewitt’, (d) edge detection ‘canny’

CHAPTER III

METHODOLOGY

This chapter will discussed about the flow chart to develop this system based on image processing, development of Graphical User Interface, system features and system's working principle. Image processing tools from MATLABR2010a is used to develop this project. Image processing is any form of signal processing for which the input is an image, such as a picture or video frame and the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Examples of image processing techniques are color thresholding, edge detection, image filtering and others.

3.1 Flow Chart

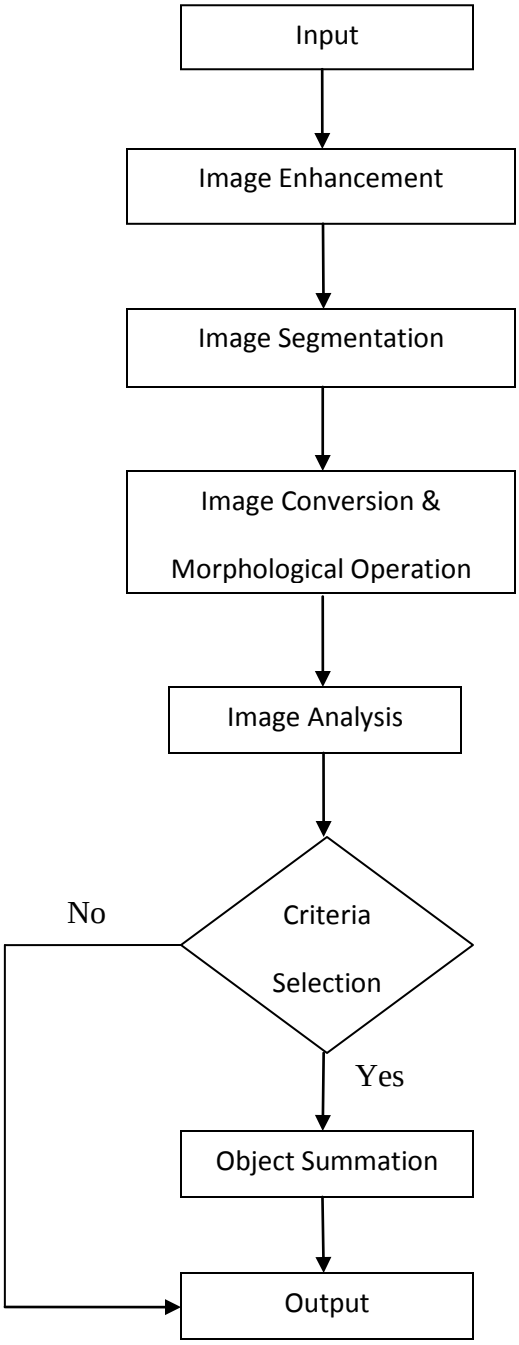


Figure 3.1 Image processing techniques perform on project

Figure 3.1 shows the flow chart of the detection and summation of pus cell used in this project. First, the sputum image is loaded. Then, image enhancement technique is performed on the image by resizing 20% of the original image. In order to separate the background with the other object, color thresholding is used. Then, image subtraction is performed where the original image is subtracted with background image to get the pus cells. Besides that, color thresholding technique is used to reduce the number of cluster while K-Means is performed later. Next, image is adjusted to obtain the contrast image followed by circular averaging filter which is used on the adjusted image. Circular averaging filter is chosen due to the objects in this image are mostly in circular shape.

Next, K-Means clustering technique is performed to separate the elements based on their intensities. Squamous epithelial cell and pus cell might be in the same cluster since the intensity of both elements is quite similar. This cluster is separated from the image for further operation. Thus, the image that is used for next operation just consists of pus cell and epithelial cell. Next, the image is converted into grayscale. So, gray thresholding technique can be performed on image which will return the image in binary form. Then, noises are removed from the image by removing objects which are more than 20 pixels.

Image analysis is done on each object. Each object is measured in terms of mean intensity, area, perimeter and centroid. It is important for criteria selection.

Criterion selection refers to the criterion that is chosen to distinguish pus cells from other objects. Criterion of mean intensity and area of pus cells are chosen for this purpose. The range of mean intensity and area of pus cells are determined. The objects which are out of this range are removed from the image. So, only pus cells will left after this process is complete.

Finally, pus cells are detected and the total number is determined. Pus cells is marked with blue line on original image. The total number of pus cells is shown through message box.

In order to make it easy to be used by user, GUI is developed to represent this project.

3.2 Design of Graphical User Interface

A Graphical User Interface (GUI) is a graphical display that contains devices or components that enable a user to perform interactive tasks. By using GUI, user does not have to type command at the command line in M-File. So, user can run the project easily by clicking some buttons on GUI since it is user friendly. For this project, the GUI consists of some push buttons, static texts, axes and panel.

The Graphical User Interface was constructed using MATLAB GUIDE or Graphical User Interface Design Environment. The GUI is designed by using layout tools provided by GUIDE. The GUI layout for this project is shown in figure 3.2.

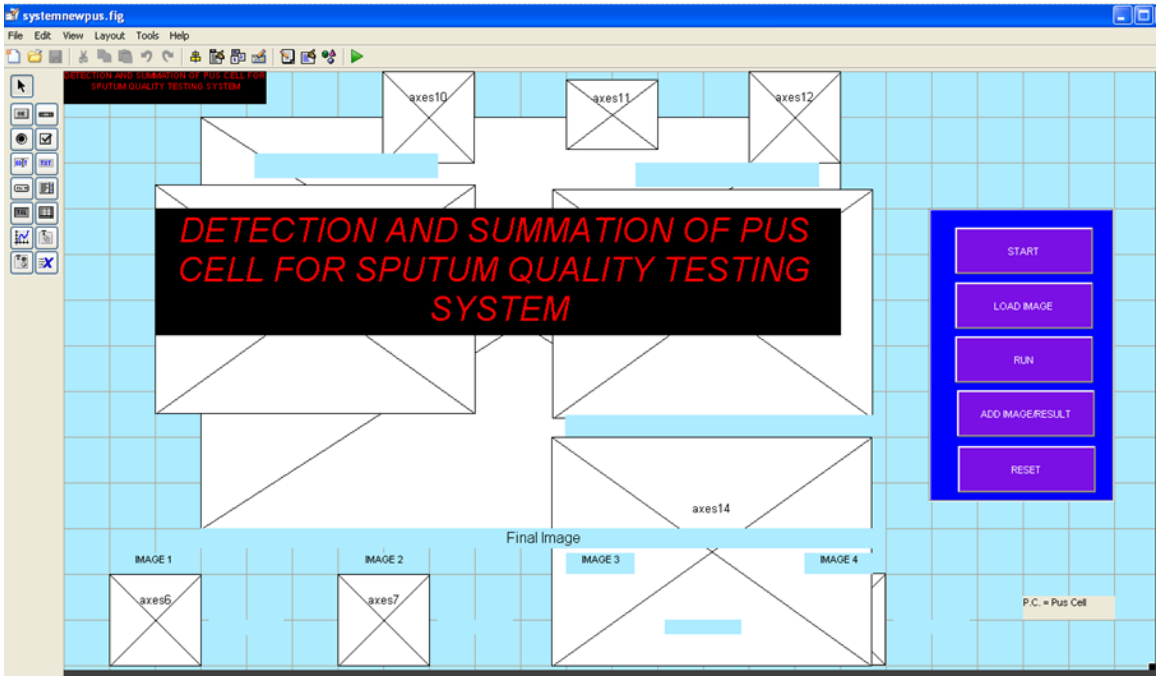


Figure 3.2 GUI Layout

Table 3.1: Tools on GUI and functions

Tools	Function
Start Button	Overall instruction of system
Load Image Button	Load image 1, image 2, image 3 and image 4
Run Button	Process the image
Add image/Result Button	Save image on the bottom of GUI/shows result
Reset Button	Reset the system
Axes 10,axes 11,axes 12	Logo of UMP,USM and VISIS
Axes 1	Original Image
Axes 2	Shows image being processed/cluster 1
Axes 14	Cluster 2
Axes 4	Final Image

Before GUI on the figure above is designed, Blank GUI (Default) template is selected and blank GUI is displayed at layout editor as shown in figure 3.3.

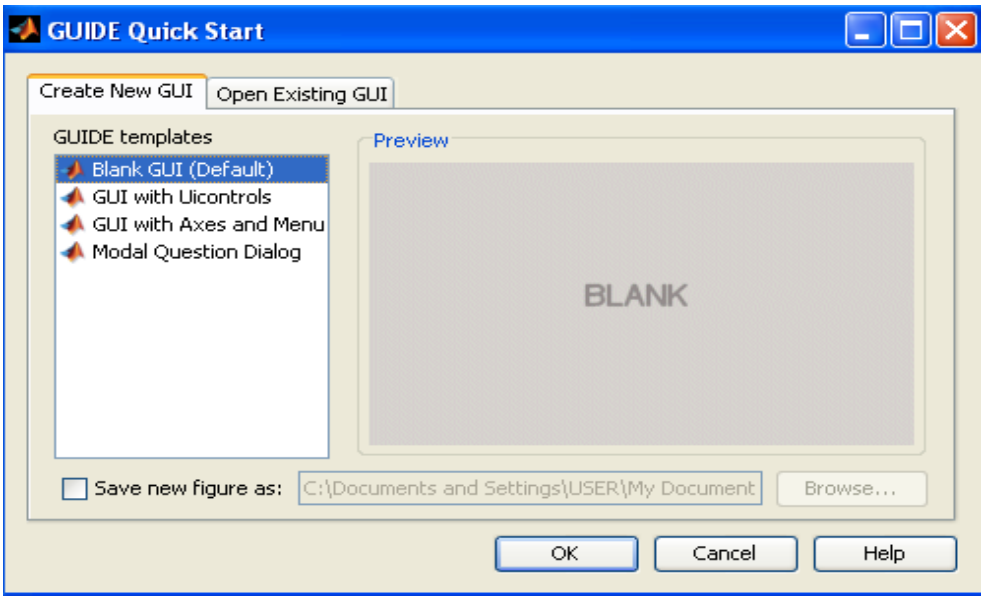


Figure 3.3 GUI Quick Start

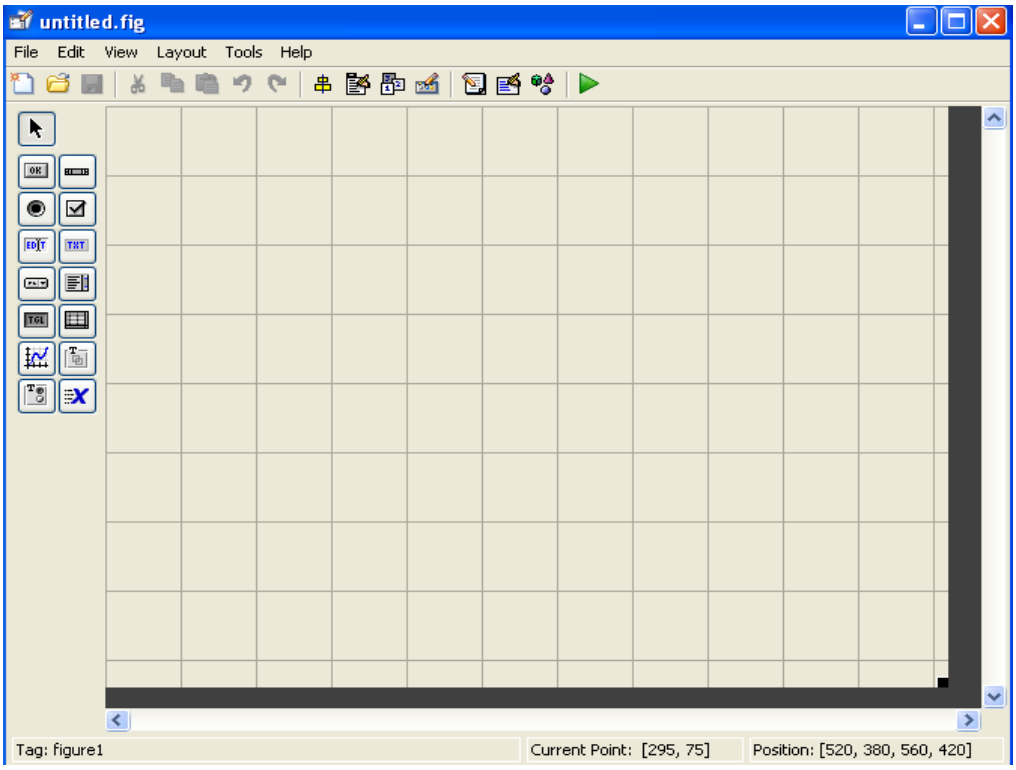


Figure 3.4 Layout Editor

3.3 System Features

This system requires 4 images that represent one sputum sample to determine the score of pus cells in that sample. Actually, these 4 images are images which are taken from 4 fields of the sputum sample. Usually, each sputum sample consists of 32 field or images and 4 images are randomly chosen from this field. The average number of pus cells is determined from these 4 images. Clear explanation regarding fields is shown through the figure 3.5.

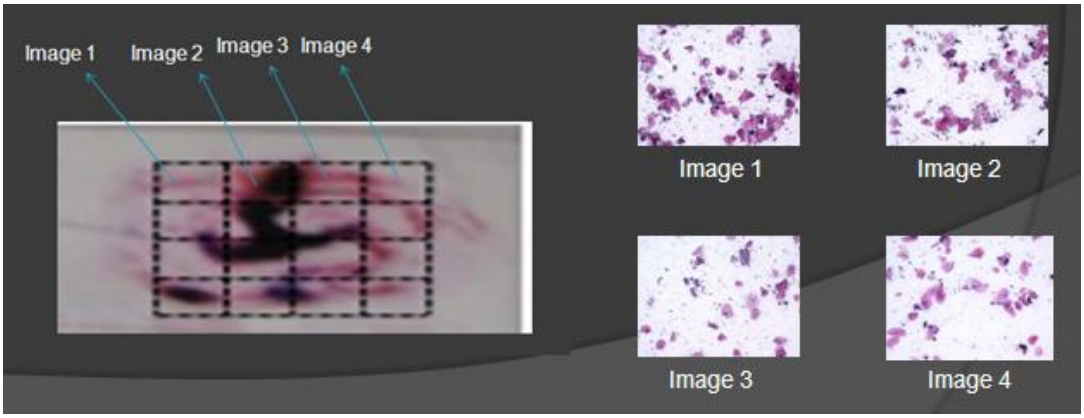


Figure 3.5 Sputum sample with 16 fields

Figure 3.5 shows the example of sputum samples with 16 fields. This is the sputum sample which has been undergone gram staining process. Gram staining is a method used to color the cells by using chloroform, iodine and other chemical reagents, so the cells can be observed clearly through microscope. The 4 images are taken from 4 different fields through digital biological microscope under x10 field. Based on the information from USM Kubang Kerian, only 4 images or fields out of 16 are required to determine the score of pus cells through the average number of pus cells from these 4 images.

3.3.1 System's working principles

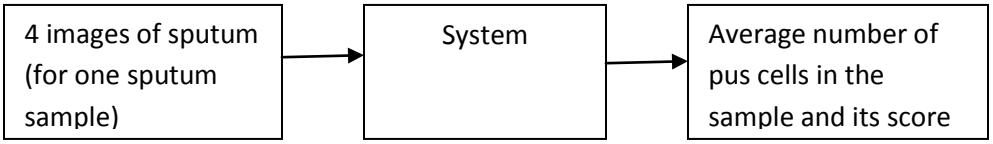


Figure 3.6 Block diagram of system's working principles

Based on the block diagram above, 4 images will be loaded one by one in the system. This system will reveal the number of pus cells in each image. At the end, the average number of pus cells in the sample is shown as well as its score.

CHAPTER IV

RESULT AND ANALYSIS

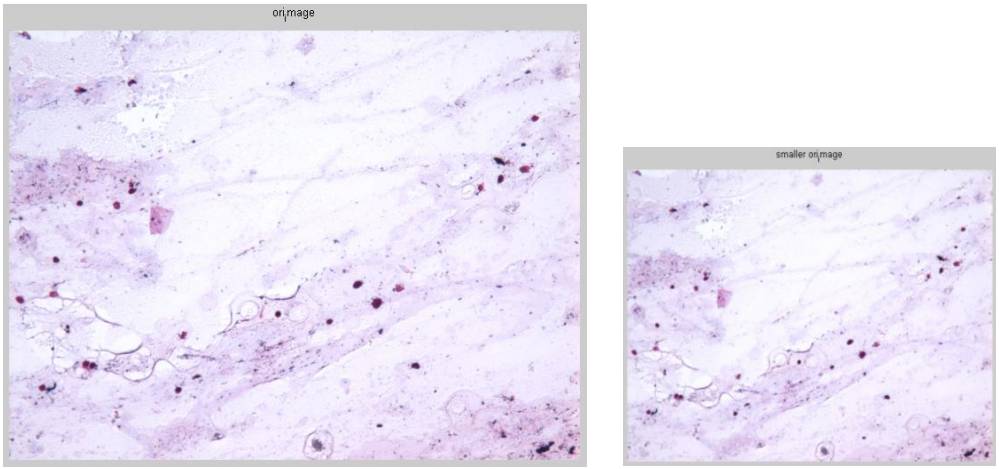
This chapter will discuss the results and analysis of image after image processing techniques were applied. The image will be in black and white colour where the white objects represent pus cells whereas the black image is a background. However, for the final image, the pus cells will be marked with blue colour on the original image. The number of pus cells in that image will be determined automatically and the result will be shown through message box.

4.1 Image Enhancement

Image enhancement is a process which will change the original image to become better image. It is used to prepare the image for the next process. This section will discuss about image resize, colour thresholding and image subtraction, image contrast and image filtering.

4.1.1 Image Resize

The original image has been resized until 20 percent from its original size. The image needs to be small in size to make sure that it can be processed faster and smoother for the next stage. The result is shown in figure 4.1.

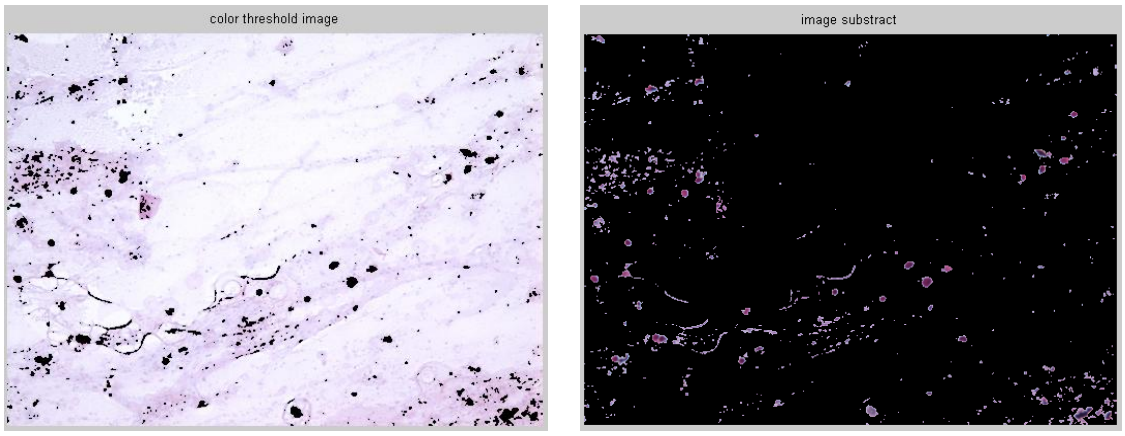


(a) (b)

Figure 4.1 (a) Original image, (b) Resize image

4.1.2 Color Thresholding and Image Subtraction

Then, color thresholding technique is performed on the resized image. Color thresholding separate the background with another object. After that, the resized image is subtracted with the threshold image. So, only another object and some noise remain in the image. The figures of image thresholding and image subtraction are shown in figure 4.2.



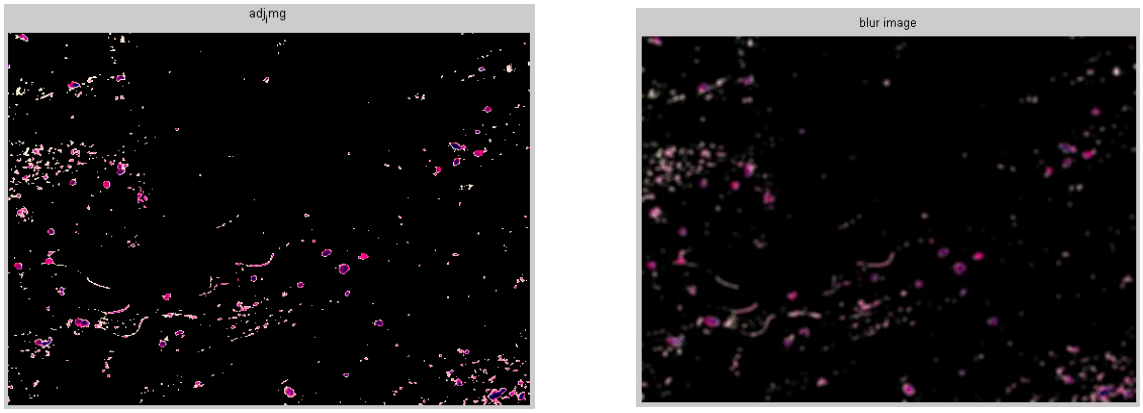
(a) (b)

Figure 4.2 (a) Color thresholding image, (b) Subtraction image

This technique should be performed to avoid too many clusters require for k-means clustering step later. Too many clusters will reduce the efficiency of the system because longer time is needed to run the system.

4.1.3 Image Contrast, Image Filtering

The previous image has been modified in terms of intensity. ‘imadjust’ command is used to contrast the image. It is important to make sure that the intensity of pus cells can be detected while k-means clustering is performed. Besides that, circular averaging filter is also applied to remove the noise resulting from k-means clustering. This filter is chosen because it can maintain the shape of pus cells which are in circular shape. The results of these two techniques are shown in Figure 4.3

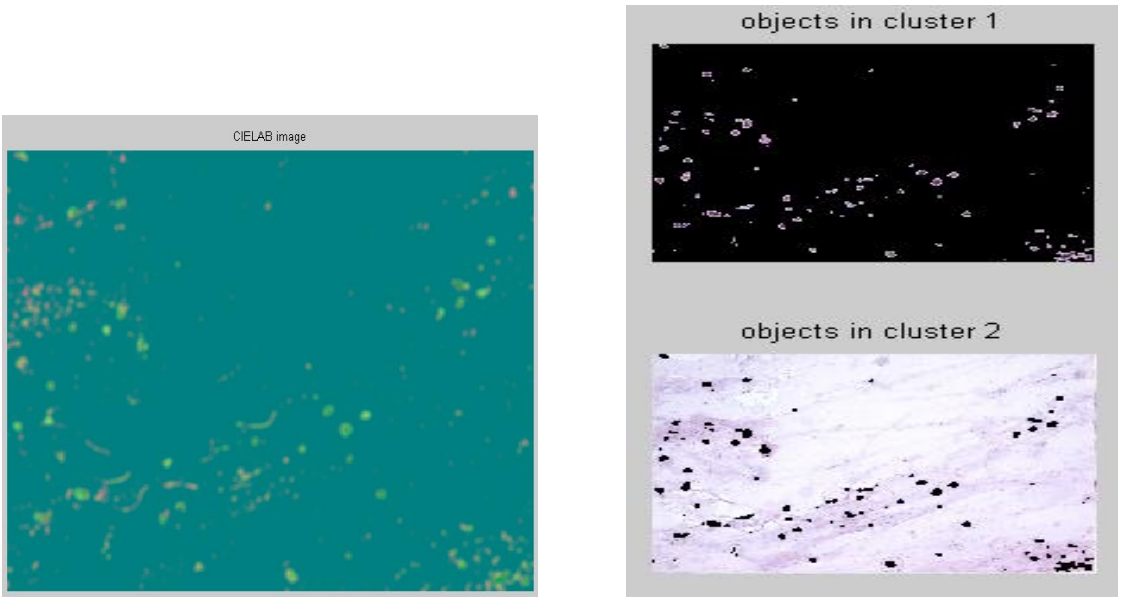


(a) (b)

Figure 4.3 (a) Contrast image, (b) Smooth or blur image (effect from circular averaging filter)

4.2 Image Segmentation

K-means clustering algorithm is chosen for this purpose. This is the technique that is usually used to separate the object into some cluster groups based on intensities. The RGB is converted into CIELAB colour space. For this pus cells image, there are three different intensities or colors that can be detected visually. So, CIELAB has ability to quantify these visual differences. Since there are two colors in the image, so the number of cluster is assigned as two. Since the cluster does not return the same intensity every time, so user needs to choose which cluster that should be proceed for the next step. The chosen cluster should contain pus cells which are in dark red colour. In this image, cluster 1 is chosen. Sometimes, cluster might consist of epithelial cells as well due to the similar intensity with pus cell. However, they can be differentiate by their sizes. It is because pus cells are smaller than epithelial cells. But, for this image, the cluster only consists of pus cells. This type of image segmentation is used based on its simplicity. The results of CIELAB color space and two clusters are shown in Figure 4.4.



(a) (b)

Figure 4.4 (a) Image in CIELAB color space, (b) Image in cluster 1 (consist pus cell and epithelial cell) and image in cluster2 (background)

4.3 Image Conversion and Morphological Operation

The chosen cluster is applied for the next steps. RGB image is converted into grayscale by using 'rgb2gray' command. It is done to prepare the image for graythresholding technique. The command of 'graythreshold' return the image in binary form which is in black and white colours. Besides that, this command can remove unwanted noises in the image. After that, 'imfill(bw,'holes')' command is applied to fill in the holes. Finally, the noises are removed by using 'bwareaopen' command. The noises which are less than 20 pixels are removed. The results for all of these image conversion and morphological operations are shown in Figure 4.5.

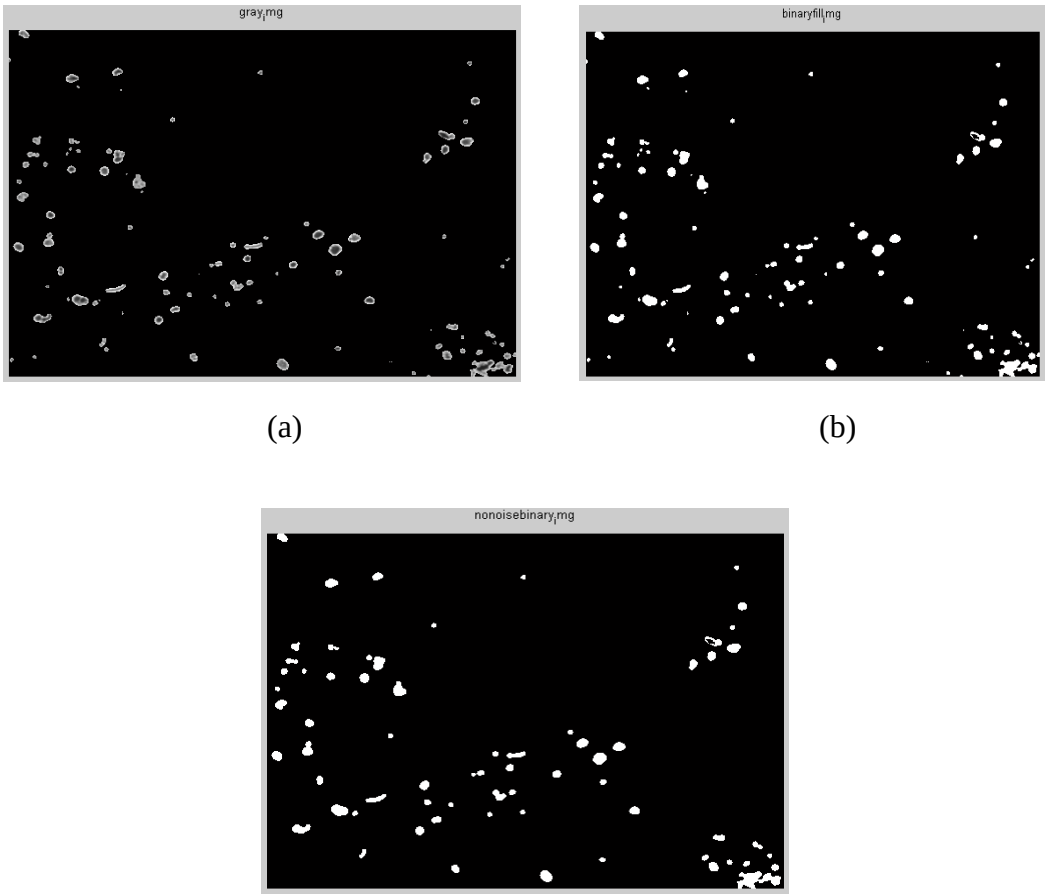


Figure 4.5 (a) Grayscale image (cluster 1), (b) Binary image, (c) Binary image (after noise have been removed)

4.4 Image Analysis

For analysis step, pus cell is be measured in terms of intensity and area. This is important to set the suitable range of intensity and area for pus cell. This range is used for criteria selection. Table 4.1 shows the mean intensity, area, perimeter and centroid of the objects in the image. Figure 4.6 shows the labeled objects with number where parameters of the objects can be determined from Table 4.1.

Table 4.1 Table from command window

Obj #	Mean Intensity	Area	Perimeter	Centroid	
# 1	157.0	88.0	33.0	10.5	242.9
# 2	161.0	21.0	14.2	11.0	169.8
# 3	166.2	90.0	34.4	14.3	186.7
# 4	168.6	74.0	30.7	15.8	4.9
# 5	170.2	40.0	21.1	17.7	150.8
# 6	173.2	57.0	31.6	25.0	139.6
# 7	168.5	59.0	28.7	29.1	123.7

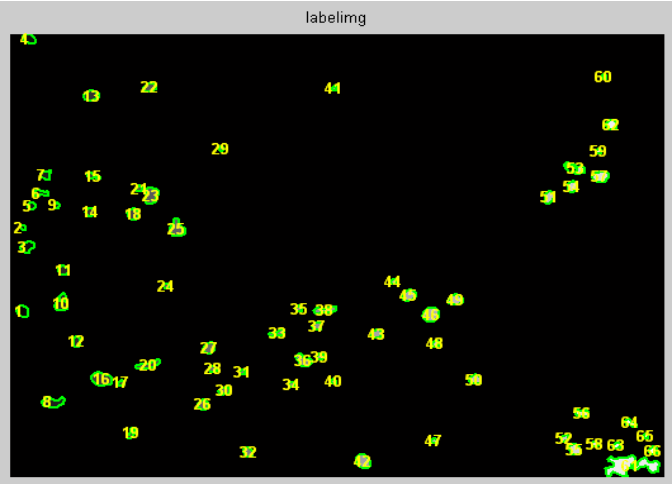
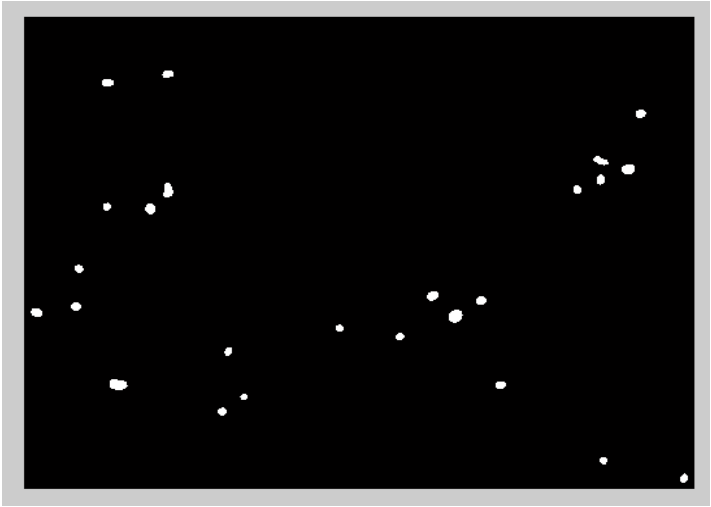


Figure 4.6 Binary image, labeled with number for each object

4.5 Criteria Selection

Three ranges which represent the intensity, area and perimeter of pus cells is used for criteria selection. The ranges are set based on the image analysis describe in subtopic 4.4. So, after some training performed on more than 20 images, the best ranges of pus cells are decided. Thus, for intensity, the value is set from 90 to 140, area is set from 20 to 100 and perimeter is set from 15 to 38. The object which is in this range is considered as pus cell and remains in this image. The other objects are eliminated.

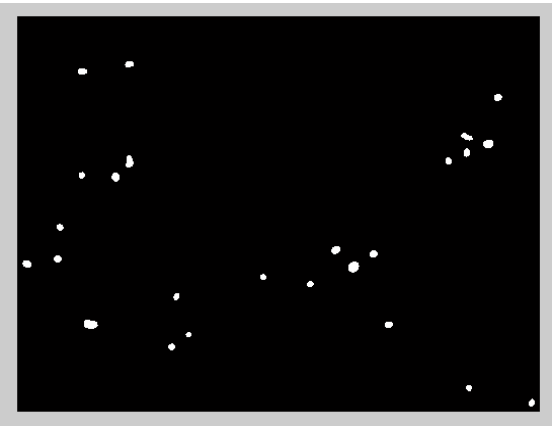


(a)

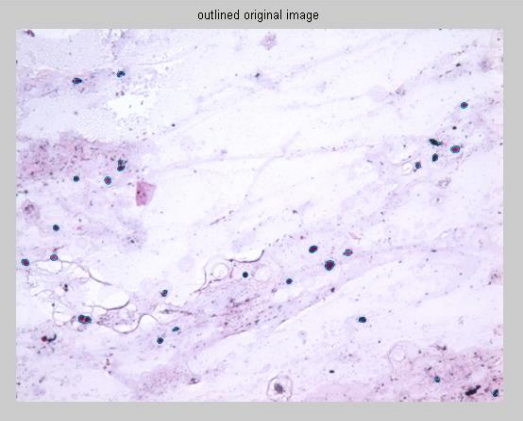
Figure 4.7 (a) Pus cells image

4.6 Image Summation

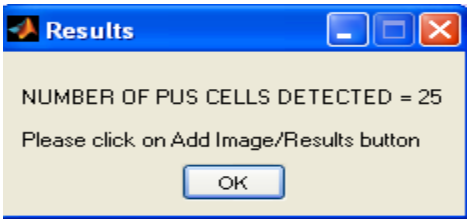
The number of pus cells from single pus cells and overlap pus cells are calculated and the total number of these cells is determined. Then, the images from single pus cells and double pus cells are added together to get the final image. 'bwperim' command is used to mark pus cells with blue colour on original image. Then, message box will appear to show the number of pus cell in the image. The results are shown in Figure 4.8.



(a)



(b)



(c)

Figure 4.8 **(**a) Total pus cells image, (b) Final image, (c) Results regarding number of pus cells in image.

4.7 Development of Automated Detection and Summation of Pus Cell for Sputum Quality Testing System using Graphical User Interface (GUI)

The method above is applied to develop system which is able to detect and count the number of pus cell for sputum quality testing. GUI is used for this purpose. Deep explanation regarding this system will be discussed in this subtopic.

4.7.1 GUI Interface

Figure 4.9 shows GUI interface for this system. This system consists of some push buttons to make the user's work become easier.

Figure 4.9 GUI interface for system.

4.7.2 System's Working Principles

4 images of sputum for each sputum sample are processed through this system. The image is loaded one by one. Determination of sputum quality testing is based on Bartlett's Criteria where it considers the number of pus cells, epithelial cells and the type of macroscopy in order to find the total score. However, this system only show the number of detected pus cells at the end of this process. After all the images have been loaded, the average number of pus cells of these 4 images is shown as a final result. Besides that, the score of pus cell for that sample is also determine by the system. The user just needs to play around with push button on the interface.

4.7.2.1 Test the System by Using Positive Sample of Sputum

For first image:

The first image is loaded by clicking the start button.

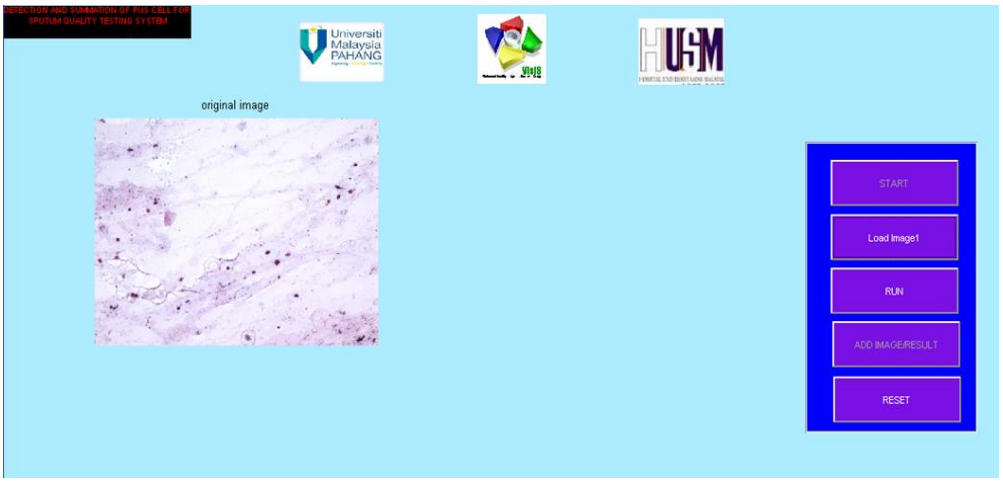


Figure 4.10 Load first image in the system.

In the middle of process, the user has to choose the cluster that needs to be processed which is the cluster that consist pus cell. This step is repeated at every loaded image. It cannot be avoided since this system use K-means clustering technique which distinguish the object based on intensity. It is not hard to make decision because usually the cluster with black background color will be chosen. Therefore, for this example, cluster 2 is chosen. User can click on cancel button to reload the image and replace the current image. It can be done if the user have loaded wrong image.

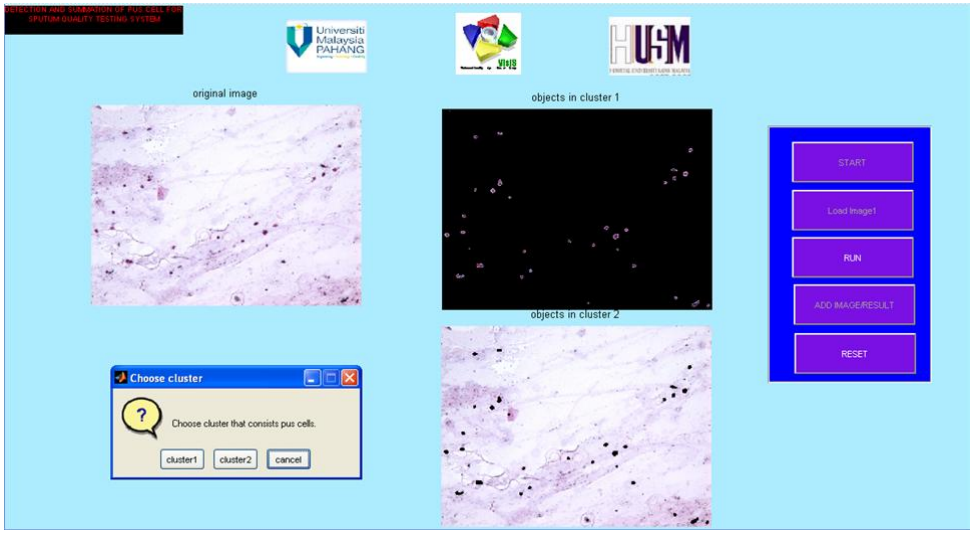


Figure 4.11 Instruction to choose cluster.

Result of first image:

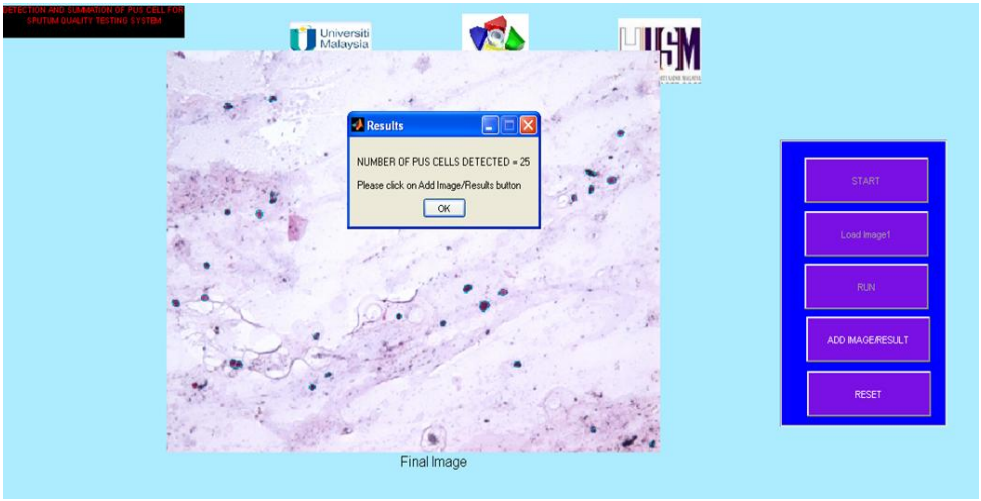


Figure 4.12 Result of first image in the system.

For the final result of first image, the image of pus cells are marked with blue line and the appeared message box shows the number of pus cells that have been detected in that image. Then, add image/result push button is clicked to save the data.

Results after clicking the add/result button:

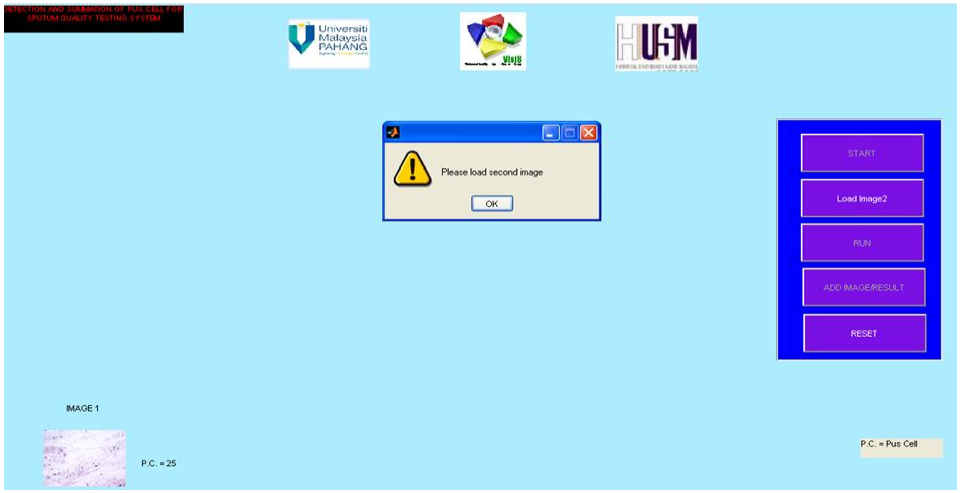


Figure 4.13 System after add/result button was clicked.

Image 1 is shown on the bottom of this interface as well as the number of pus cells .Then, next instruction is shown through message box.

For second image:

The steps of second image and another 2 image are same with the first image.

Load second image:

Figure 4.14 Load second image in the system.

Result of second image:

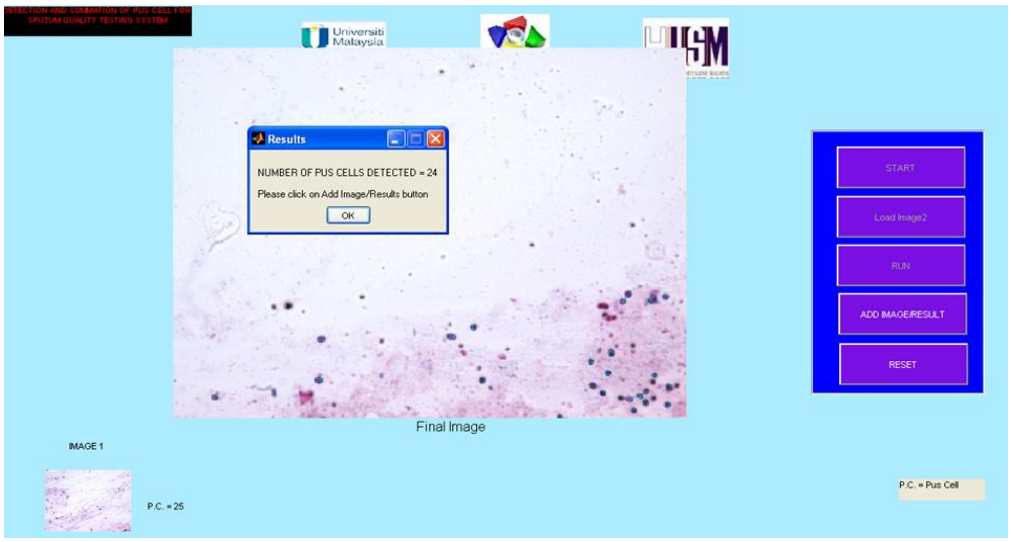


Figure 4.15 Result of second image in the system.

Results after clicking the add/result button:

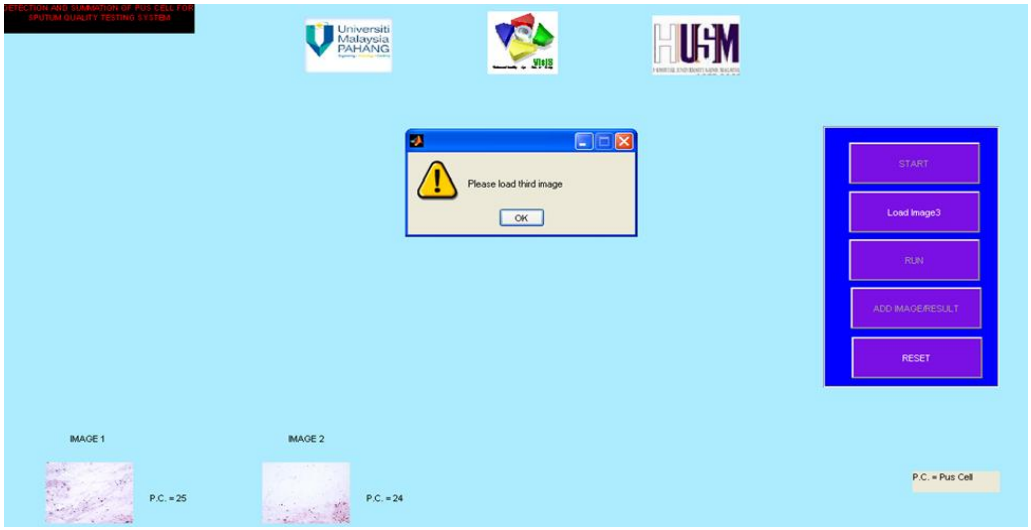


Figure 4.16 System after add/result button was clicked.

Load third image:

Figure 4.17 Load third image in the system.

Result of third image:

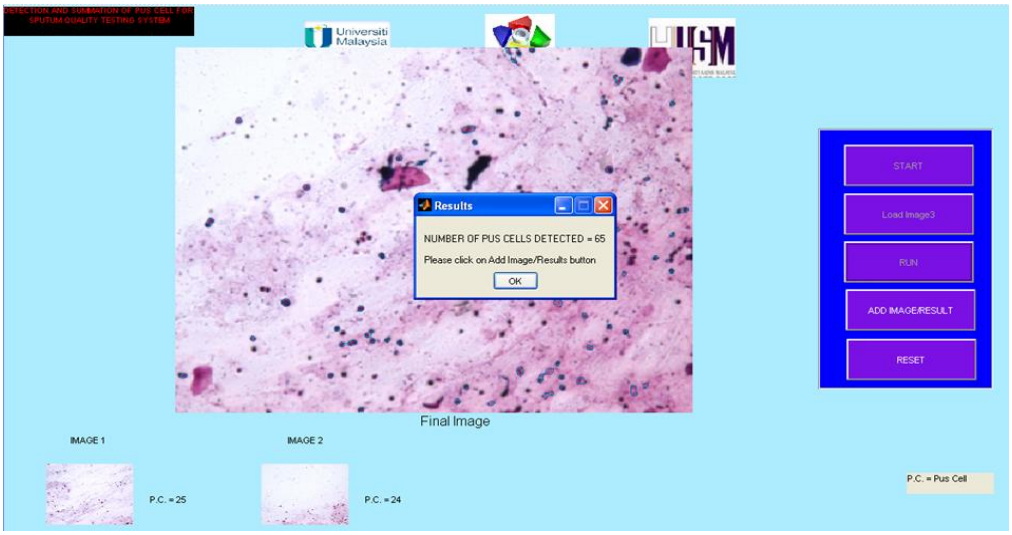


Figure 4.18 Result of third image in the system.

Results after clicking the add/result button:

Figure 4.19 System after add/result button was clicked.

Load fourth image:

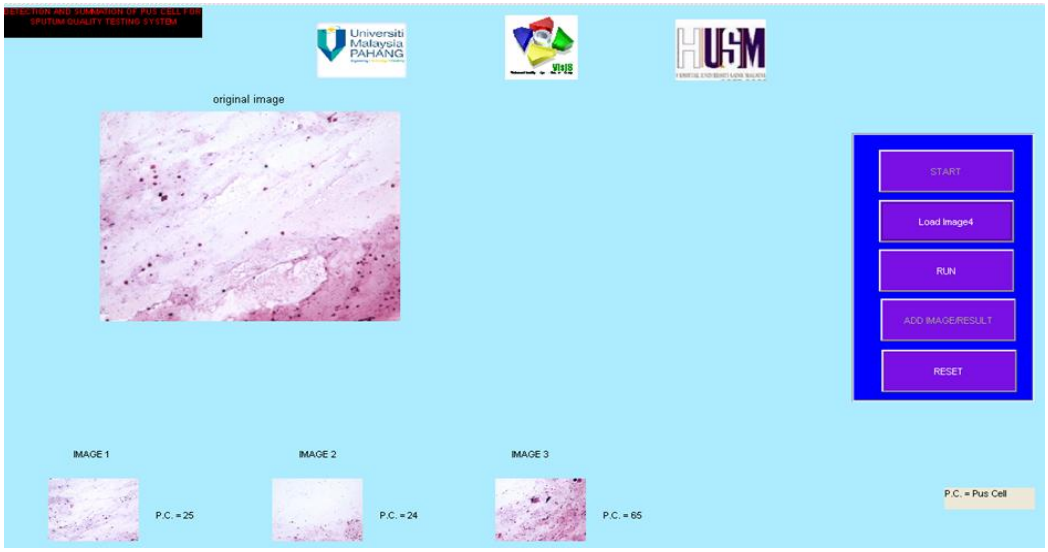


Figure 4.20 Load fourth image in system.

Result of fourth image:

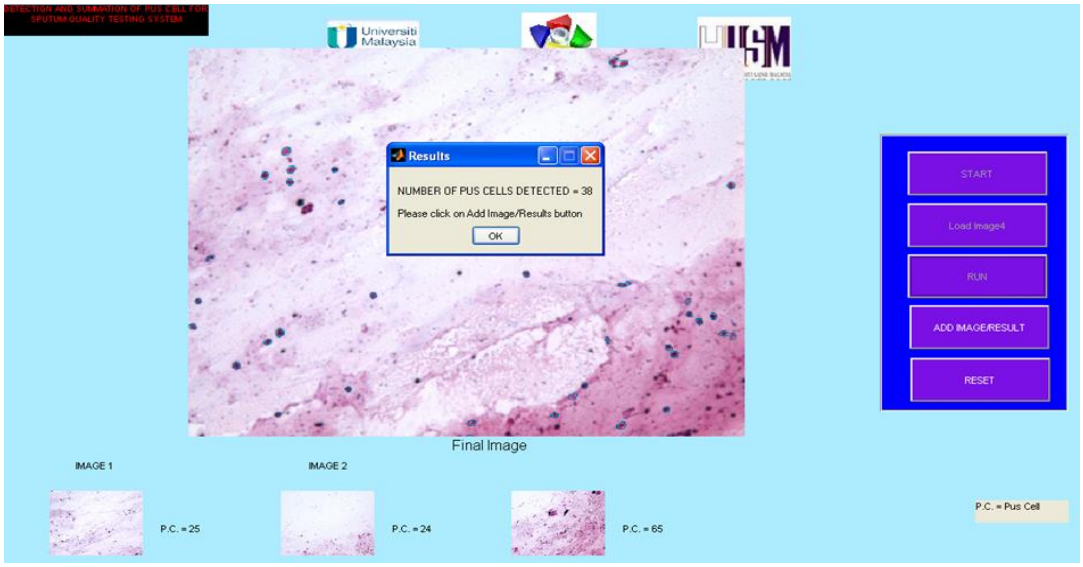


Figure 4.21 Result of fourth image in system.

Results after clicking the add/result button:

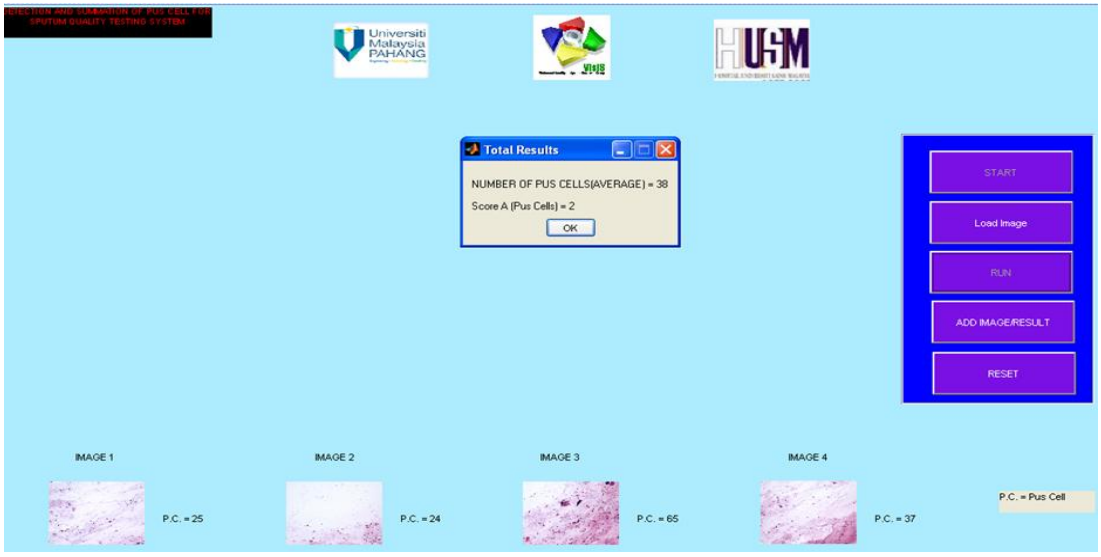


Figure 4.22 Final result of the system.

The average number of pus cells from these 4 images is shown as the final result. Besides that, the score for pus cell which is based on Bartlett's Criteria are shown as well. The result shows that the average number of pus cell is 38 and the score for pus cell based on Modified Bartlett's Criteria is +2.

4.7.2.2 Test the System by Using Negative Sample of Sputum

The steps are exactly same as the steps shown above. The final results of negative sample of sputum:

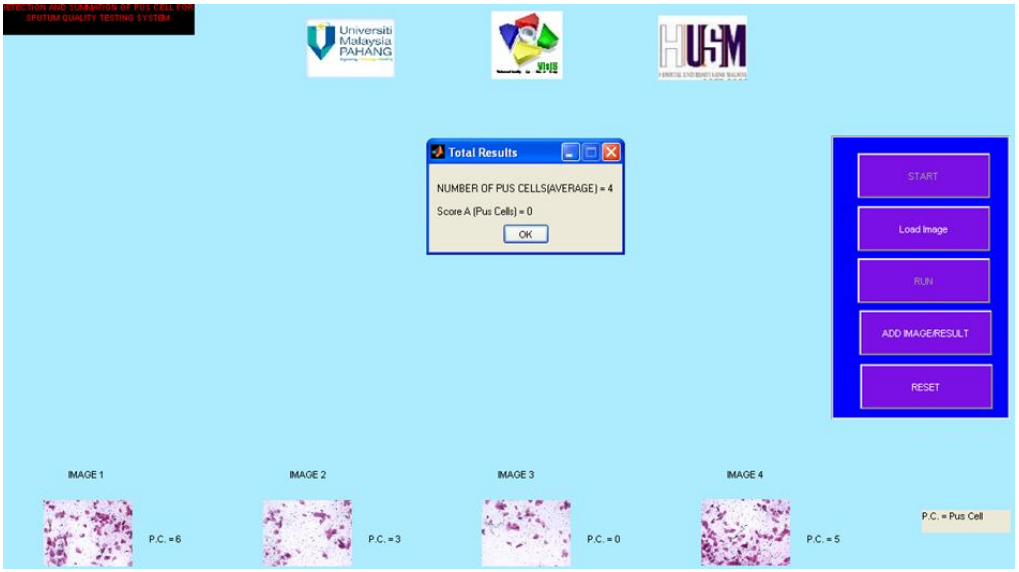


Figure 4.23 Final result of the system

The result shows that the average number of pus cell is 4 and the score for pus cell based on Modified Bartlett's Criteria is 0.

CHAPTER V

DISCUSSION

Without this system, the detection and summation of pus cells to determine the quality of sputum based on Bartlett's criteria are done manually by human. In manual method, the score is determined based on the estimation number of cells (in range). By using this system, the detection and summation of pus cells can be determined automatically through computer. The number of cell is counted one by one. User especially technologist can play around with push buttons on the GUI to evaluate the sample of sputum. This system is more effective since it takes 30 seconds for each image and less than 3 minutes for one sample compare to manual method which will takes longer time to make the decision. Furthermore, this system is based on the application of technology where image processing techniques are used to develop this system and this system can be applied on computer.

5.1 Validation Test

Validation test is the comparison between number of pus cell that have been determined by the system and number of pus cell calculated by technologist in USM Kubang Kerian. Validation test is also done based on the score of pus cell for each sample. Validation test was carried out to determine the effectiveness of this system.

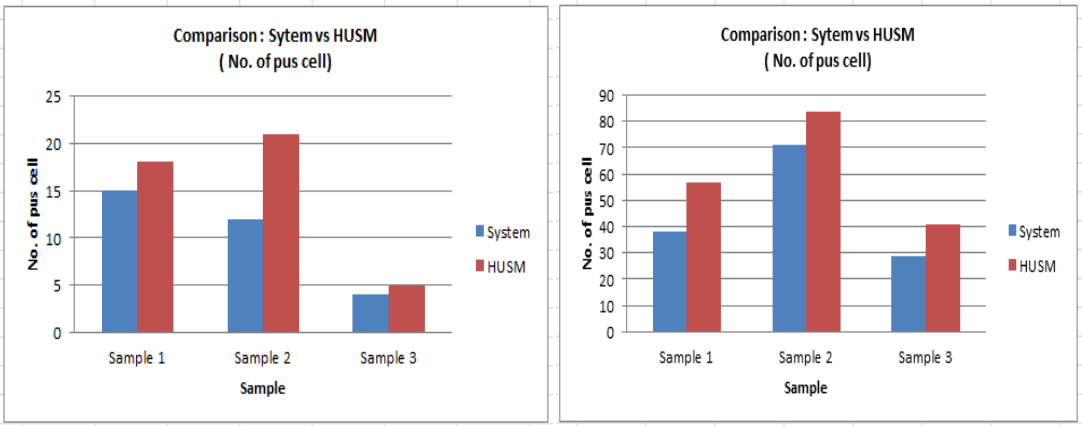
This system cannot detect all of the number of pus cells in image perfectly. However, the number of detected cells always in the range which is valid based on Bartlett's

criteria. For example, by referring to Appendix A of the image 4 in sample 1 (positive sample), the number of pus cells is found to be 38 when manual method is applied which is the exact number of pus cells in image. But, by using this system, the number of pus cells is recorded as 49. So, eventhough there is slightly different compare to exact number, but based on Bartlett's criteria, 38 and 49 are larger than 25 where both of these value will have the score of +2. In addition, if error occurs in one image, there is another 3 images that needs to be processed where the total score is determined based on average number of pus cells and epithelial cells from four images. So, the error can be reduced through the average calculation. The tendency of error on evaluating the sputum sample can be decreased as well.

This subtopic will roughly discuss about the validation test result based on the results that have been summarized in the form of graphs. However, the detail results regarding this validation test can be obtained by referring to Appendix A.

5.1.1 Number of Pus Cell and Score for Each Sample

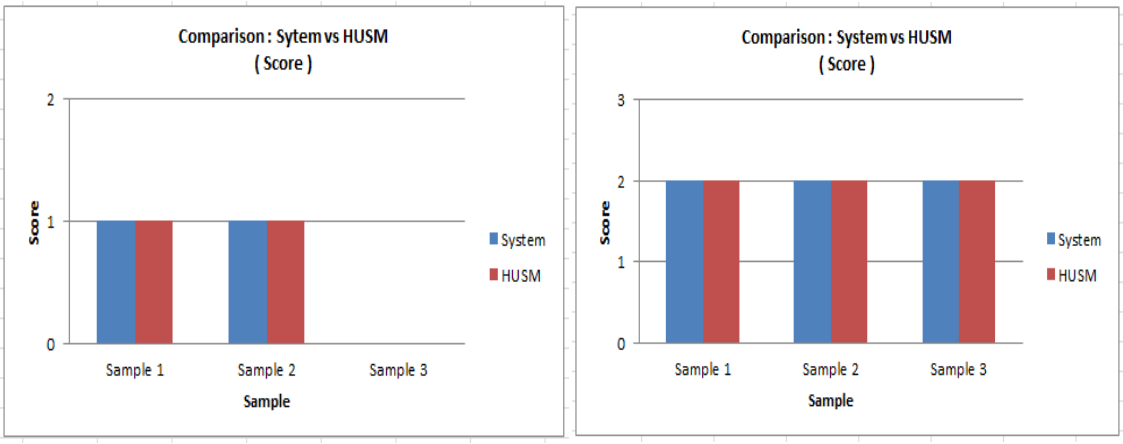
The comparison between average number of pus cell as well as score for each sample which is determined from the graph and from technologist is shown through the graphs in Figure 5.1. The samples are categorized as positive sample and negative sample. Each sample, for example sample 1, there are 4 images that have been evaluated and the final result (average number of pus cell) is determined based on average calculation of number of pus cell from these 4 images. Then, the score is determined from the average number of pus cell by referring to the Bartlett's Criteria. For analysis purpose, 3 samples of sputum have been tested for each positive and negative sample.



(a) (b)

Figure 5.1 (a) Comparison between system and HUSM in terms of average number of pus cell for negative sample, (b) Comparison between system and HUSM in terms of average number of pus cell for positive sample.

The graphs in Figure 5.1 show the comparison or difference of average number of pus cell which is determined from the system and from USM's technologist. Based on the graphs, for both negative and positive samples, there is slightly difference between the number of pus cell evaluated by the system and by the technologist. For negative sample, the largest difference comes from sample 2 with the difference of 9 and for positive sample, the largest difference comes from sample 1 with the difference of 19. Then, the smallest difference for negative sample is 1 which is from sample 3 and the smallest difference for positive sample is 12 which is from sample 3.



(a) (b)

Figure 5.2 (a) Comparison between system and HUSM in terms of score for negative sample (pus cell), (b) Comparison between system and HUSM in terms of score for positive sample (pus cell).

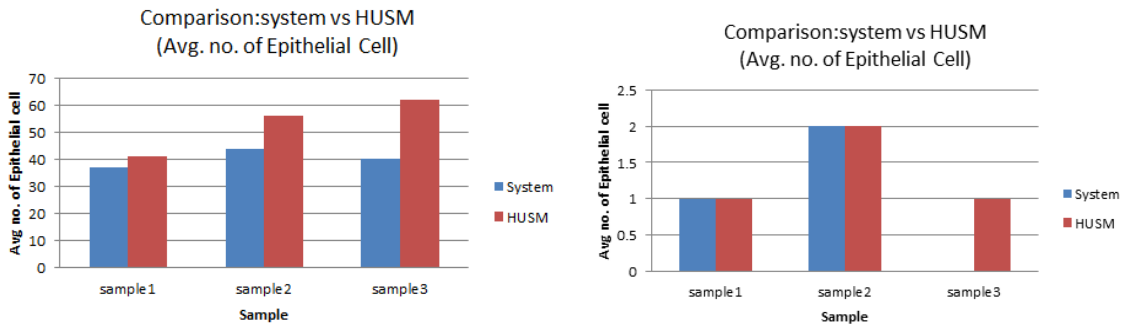
Based on the graphs in Figure 5.2, the comparison is based on score. For negative samples, the score for sample 1 and 2 are evaluated as +1 and score for sample 3 is zero. For positive samples, all of the three samples have +2 score. The graphs also show that the same scores are determined by the system and by USM's technologist for all of the samples.

Thus, based the graphs in Figure 5.1 (average number of pus cell), this system cannot detect and count all of the pus cells in image perfectly. Actually, error of detection comes from some factor which will be discussed in subtopic 5.2. However, the number of detected cells always in the range which is valid based on Bartlett's criteria. So, the score for pus cell evaluated using system and by USM's technologist should be the same since the score is determined based on the range of number of pus cell. Actually, one of the elements to determine the quality of sputum by using Bartlett's Criteria is based on the score of pus cell. So, the score result is more important than the average number of pus cell.

In conclusion, based on the score result, this system is efficient to be applied since the scores determined from the system and from USM's technologist are same for all of the 6 tested samples.

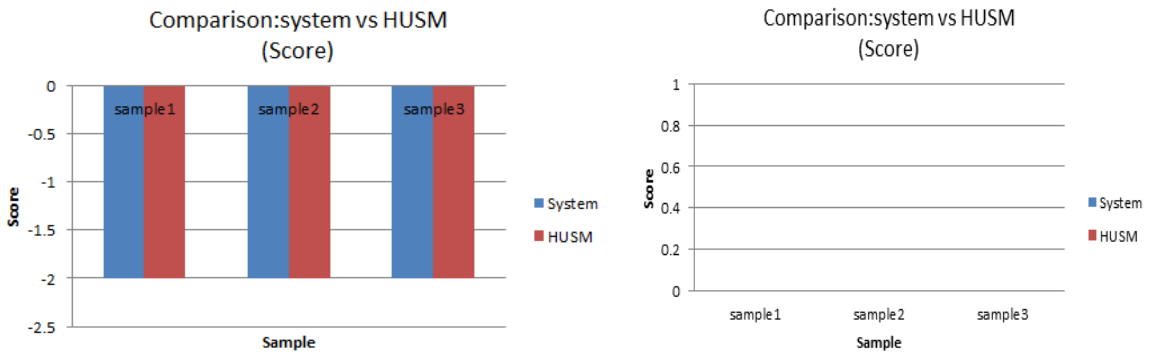
5.1.2 Validation test result (pus cell and epithelial cell)

Extra works have been done to develop a complete system which is able to detect pus cell and epithelial cell based on Bartlett's criteria. The coding to detect and count the number of epithelial cell is developed and combined with the coding to detect and count the number of pus cell. This complete system also reveals the score for pus cell and epithelial cell of sputum sample. At the end, it shows the quality result of the sputum either positive or negative. Actually, this complete system is the extension from this final year project and it is developed because the quality of sputum cannot be determined just based on pus cell but the results from epithelial cell is also requires in order to fulfill the Bartlett's criteria. The validation test has been done and the full result is shown in Appendix A. Some analysis based on this validation result is discussed in this subtopic.



(a) (b)

Figure 5.3 (a) Comparison between system and HUSM in terms of average number of epithelial cell for negative sample, (b) Comparison between system and HUSM in terms of average number of epithelial cell for positive sample.



(a) (b)

Figure 5.4 (a) Comparison between system and HUSM in terms of score for negative sample (epithelial cell), (b) Comparison between system and HUSM in terms of score for positive sample (epithelial cell).

The type of sample used to detect epithelial cell is the same with the sample used to detect pus cell. Based on the graphs in figure 5.3 and figure 5.4, just like validation test result which has been discussed in subtopic 5.1.1, there is slightly difference regarding the average number of pus cell and epithelial cell evaluated by the system and by USM's technologist. However, the score is the same for both methods. The score of epithelial cell for negative sample is -2 whereas the score of epithelial cell for positive sample is 0. Thus, this system is efficient to be applied to determine the quality of sputum. The difference number of pus cell and epithelial cell which is evaluated by system and USM's technologist for each image is summarized in Table 5.1 and Table 5.2.

Table 5.1 Difference number of cell (comparison between system and manual) for negative sample

Sample (Negative)	Difference number of cells									
	Image 1		Image 2		Image 3		Image 4		Average	
	Pus	Epi	Pus	Epi	Pus	Epi	Pus	Epi	Pus	Epi
1	6	4	0	8	2	0	3	3	3	4
2	20	14	6	13	3	0	5	21	9	12
3	3	31	2	16	1	14	4	26	1	22

Table 5.2 Difference number of cell (comparison between system and manual) for positive sample

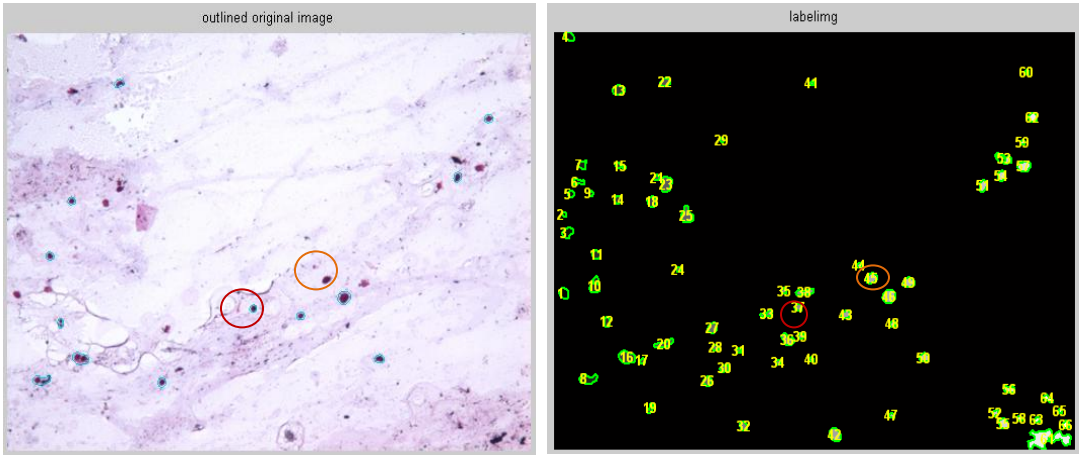
Sample (positive)	Difference number of cells									
	Image 1		Image 2		Image 3		Image 4		Average	
	Pus	Epi	Pus	Epi	Pus	Epi	Pus	Epi	Pus	Epi
1	1	1	11	0	56	1	11	0	19	0
2	29	1	5	1	9	1	18	2	13	0
3	6	1	15	1	16	0	14	0	12	1

5.2 Factors of Error in Detection.

The factors which cause the difference number of pus cell determined from the system and from USM's technologist will be discussed in this subtopic. There are two factors that lead to this problem which are missed detection and wrong detection. These factors come from the system itself which still needs to be improved. However, sometimes the image like blur image also contributes to this problem.

5.2.1 Missed Detection

There is some situation where the software supposed to detect pus cell in image but it failed to do so. There are some factors that lead to this problem. The main factor is due to the object or pus cell which is out of the specific range of mean intensity, area and perimeter that has been set to detect pus cell.



(a) (b)

Figure 5.5 (a) Original image, (b) Image labeled with number.

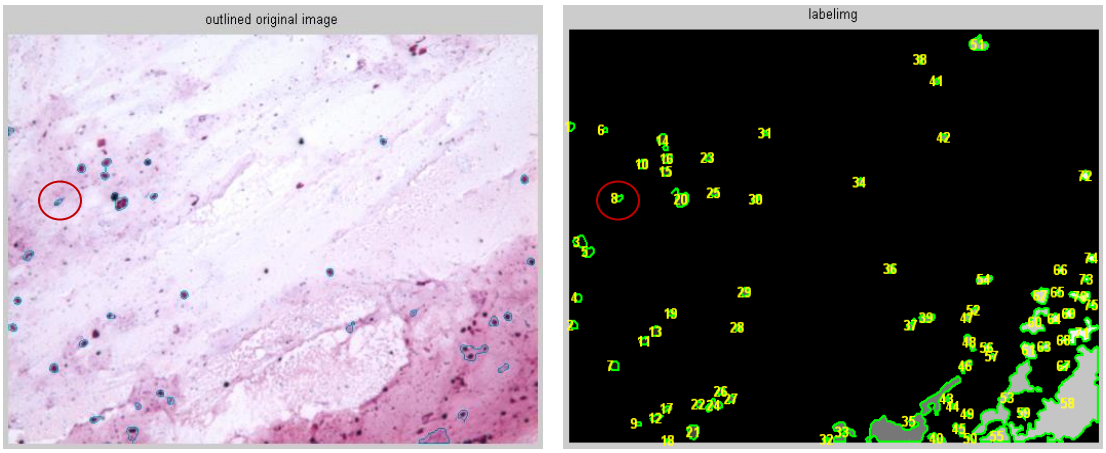
Table 5.3 Parameters of all objects in image.

Obj #	Mean Intensity	Area	Perimeter	Centroid		Diameter
# 1	157.0	88.0	33.0	10.5	242.9	10.6
# 2	161.0	21.0	14.2	11.0	169.8	5.2
# 3	166.2	90.0	34.4	14.3	186.7	10.7
# 4	168.6	74.0	30.7	15.8	4.9	9.7
# 5	170.2	40.0	21.1	17.7	150.8	7.1
# 6	173.2	57.0	31.6	25.0	139.6	8.5
# 7	168.5	59.0	28.7	29.1	123.7	8.7
# 8	143.2	135.0	50.0	34.4	322.4	13.1
# 9	164.3	21.0	14.5	37.6	150.5	5.2
#36	176.7	98.0	40.6	232.0	285.9	11.2
#37	148.0	50.0	23.9	242.9	255.8	8.0
#38	190.4	90.0	45.0	249.3	241.9	10.7
#39	174.7	34.0	19.9	245.4	282.5	6.6
#40	175.4	22.0	15.1	255.7	304.1	5.3
#41	166.2	20.0	14.2	256.4	48.0	5.0
#42	157.0	122.0	39.2	279.2	374.0	12.5
#43	161.8	59.0	25.9	289.8	262.7	8.7
#44	188.8	24.0	15.7	303.3	217.0	5.5
#45	149.4	95.0	34.4	315.3	229.0	11.0
#46	141.2	139.0	40.6	332.6	245.7	13.3
#47	153.8	26.0	16.5	335.3	356.1	5.8

For example, based on the table 5.3 the pus cell in red circle can be detected by the system compare to the pus cell in brown circle. Based on figure 5.3, the pus cell in red circle is labeled with 37 whereas the pus cell in brown circle is labeled with 45. The table in table 5.3 shows the value of mean intensity, area, perimeter, centroid and diameter of in image. In programming part, detection of pus cell is based on intensity and area. Thus, the most suitable range of intensity of pus cell that has been decided for this project is 120 – 163 whereas the range for area is 35 – 81. The objects which are out of these two ranges are not pus cell. For object with labeled 37, the intensity is 148 whereas the area is 50. So, it is considered as pus cell since the value of intensity and area are still in the range of pus cell. However, for object with labeled 45, it is not consider as pus cell because the intensity is 149.4 and the area is 95. Eventhough the intensity is still in the range of pus cell, but the area is totally out from area range. The coding has been set that the object must be in the range of both intensity and area to be considered as pus cell.

5.2.2 Wrong detection

Sometimes, this system will detect object which is not pus cell. The same factor as discuss in subtopic 5.2.1 lead to this problem.



(a) (b)

Figure 5.6 (a) Original image, (b) Image labeled with number.

Table 5.4 Parameters of all objects in image.

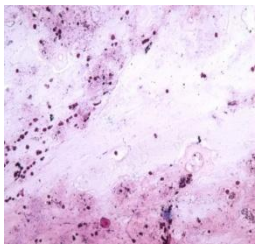
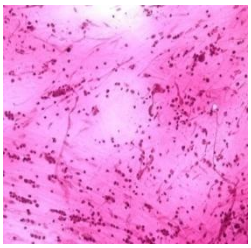
Obj #	Mean Intensity	Area	Perimeter	Centroid		Diameter
# 1	161.0	39.0	22.5	2.9	91.7	7.0
# 2	148.4	52.0	23.9	5.5	277.7	8.1
# 3	153.6	119.0	39.8	11.7	199.7	12.3
# 4	142.1	48.0	23.3	9.6	252.4	7.8
# 5	144.1	77.0	31.6	19.9	209.5	9.9
# 6	160.1	21.0	13.7	36.0	95.0	5.2
# 7	130.3	70.0	27.9	45.2	316.1	9.4
# 8	155.1	38.0	21.9	49.3	159.0	7.0
# 9	163.1	25.0	16.2	68.1	370.4	5.6

Based on the table 5.4, the object in red circle is not pus cell but this system still considers it as pus cell. This object which is labeled as 8 has mean intensity of 155.1 and area 38 as stated in the table in table 5.4. These two values are in the range of intensity and area of pus cell. That is the reason why this object is detected eventhough it is not pus cell. So, this system sometime creates an error due to this wrong detection.

5.3 System Limitation

The detection of pus cells and epithelial cells is based on intensity of object. Therefore, there is certain image that can be well processed by this system and gives accurate result. The deep explanation regarding limitation based on image is summarized in table 5.5.

Table 5.5 Factors that affect the result of the system

Factor	Result from system	
	Good	Worse
Background of image	 This image gives better result because the image is smooth and clear.	 This image gives worse result because the background is too bright (pink) and blur
Dimension of image (JPEG)	2576 x 1932	Other than 2576 x 1932

5.4 Comparison Between Manual Method and System

Detection and summation of pus cells by using system based on image processing gives more benefits than manual method conducted by human. The comparison between manual method and detection using system is summarized through the Table 5.6.

Table 5.6 Comparison between manual method and system

Manual method	Software
Requires long time period to calculate number of pus cells	Requires short time period to calculate number of pus cells
Only technologist can perform this task.	Other staff can performed this task if the technologist absent.
Conditions of technologist such as emotion sometimes affect the results.	Results are not affected by emotion.
Easy to get bored, tired etc.	Easy to use /User friendly.
Requires concentration when counting the cells.	Can do another job while waiting for the result.

CHAPTER VI

CONCLUSION

6.1 Conclusion

This system can detect and count the number of pus cells in sputum image. It can also determine the score for pus cells of sputum sample based on Modified Bartlett Criteria. The score is determined based on the range of number of pus cells. There is slightly different regarding the number of pus cells calculated using manual method and using the system. However, this error does not affect the score of pus cells which is the main factor to determine the quality of sputum. So, this system still valid to determine the score for pus cells based on Bartlett Criteria. Besides, this system gives a lot of benefits especially in medical field. Manual method can be eliminated and replace with automated detection and summation of pus cell system which is better than method conducted by human.

6.2 Future Development

For future development, this vision system can be connected with the hardware like digital biological microscope where the online image can be directly processed by the vision system to get the result. In other words, the image can be captured in real time. So, this new system can evaluate the sample in shorter time.

REFERENCES

1. Zakuan Z. Deris, Siti Asma' Hassan, Zeehaida Mohamed, Habsah Hassan, "Implementation of routine sputum rejection criteria to improve the outcome of sputum culture result", International medical journal vol. 15, no. 4, pp. 287-290, September 2008.

2. Suman Tatiraju, Avi Mehta: Image segmentation using k-means clustering, EM and Normalized cuts.

3. S. Anitha Elavarasi, Dr. J. Akilandeswari, Dr. B. Sathiyabhama, "A survey on partition clustering algorithm", International Journal of Enterprise Computing and Business Systems Vol. 1 Issue 1 January 2011.

4. H.P. Ng, S.H. Ong, K.W.C. Foong, P.S. Goh, W.L. Nowinski: Medical image segmentation using k-means clustering and improved watershed algorithm (2006).

5. Federico Bussolino, Marina Sironill, Elena Bocchietto , and Alberto Mantovani, "Synthesis of Platelet-activating Factor by Polymorphonuclear Neutrophils Stimulated with Interleukin-8", The journal of biomedical chemistry vol. 267, no. 21, issue of July 25, pp 14598-14603, 1992.

6. Morisuke Otani: The significance of the epithelial cells and saprophytes in sputum (1916).

7. Dalong Li, Russell M. Mersereau, Steven Simske: Blur identification based on Kurtosis minimization.

8. Matlab R2010a: Matlab help, fspecial: circular averaging filter.

9. Rafael C. Gonzales, Richard E. Woods, Digital Image Processing, 3rd Edition, Pearson Prentice Hall.

10. Huiyu Zhou, Jiahua Wu & Jianguo Zhang, Digital Image Processing Part II, Ventus Publishing APs, 2010.

11. Lester K. Wong, Arthur L. Barry, Susan M. Horgan," Comparison of Six Different Criteria for Judging the Acceptability of Sputum Specimens", Journal of clinical microbiology, Oct. 1982, vol. 16, no. 4, p. 627-631.

12. Morten Alhede , Thomas Bjarnsholt , Peter . Jensen, Richard Kerry Phipps, Claus Moser, Lars Christophersen, Louise Dahl Christensen, Maria van Gennip, Matt Parsek, Niels Hoiby, Thomas Bovbjerg Rasmussen and Michael Givsko, *Pseudomonas aeruginosa* recognizes and responds aggressively to the presence of polymorphonuclear leukocytes(2007).

13. Kamarul Hawari, Nurul Wahidah, Faradila, Rosyati, Nazriyah, Rohana, Farizan, Falfazli, Lailatul, The Computer Vision System Module, 1st Edition.

Appendix A

Validation Test Result for Pus cell

Negative Sample

i. Sample 1

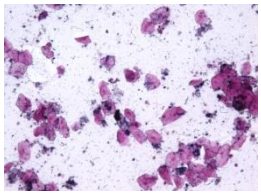
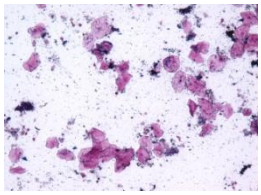
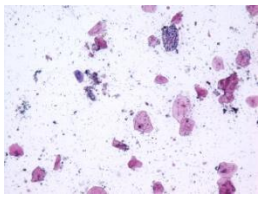
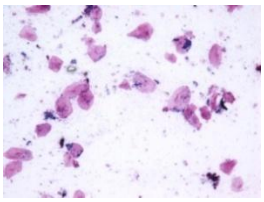





Image 1 Image 2 Image 3 Image 4

Pus cell

Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
<10 , score:0							X (6)	X 9		
10-25, score:+1	X (14)	X 20	X (24)	X 24	X (15)	X 17			X (15)	X 18
>25, score:+2										

Epithelial Cell

Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
<10 , score:0										
10-25, score: -1										
>25, score: -2	X (56)	X 60	X (36)	X 44	X (29)	X 29	X (27)	X 30	X (37)	X 41

Result : Negative / Negative

ii. Sample 2

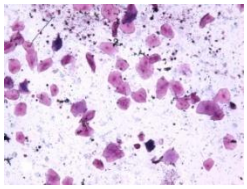
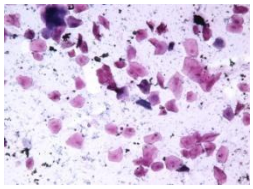
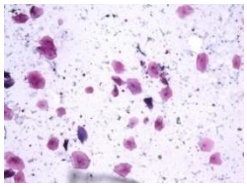
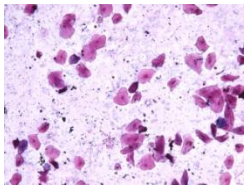





Image 1 Image 2 Image 3 Image 4

Pus cell

Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
<10 , score:0			X (6)				X (2)	X 7		
10-25, score:+1	X (25)			X 12	X (15)	X 18			X (12)	X 21
>25, score:+2		X 45								

Epithelial cell

Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
<10 , score:0										
10-25, score: -1										
>25, score: -2	X (39)	X 53	X (59)	X 72	X (32)	X 32	X (46)	X 67	X (44)	X 56

Result : Negative / Negative

iii. Sample 3

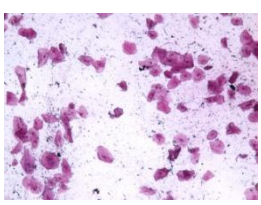
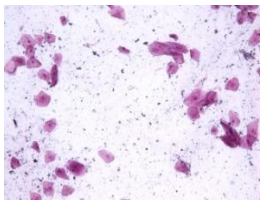
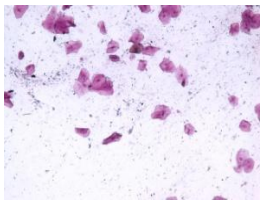
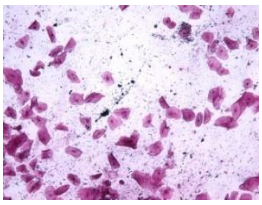





Image 1 Image 2 Image 3 Image 4

Pus cell

Cells	Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
		Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
Pus	<10 , score:0	X (6)	X 3	X (3)	X 5	X (0)	X 1	X (5)	X 9	X (4)	X 5
	10-25, score:+1										
	>25, score:+2										

Epithelial cell

Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
<10 , score:0										
10-25, score: -1										
>25, score: -2	X (49)	X 80	X (29)	X 45	X (23)	X 37	X (60)	X 86	X (40)	X 62

Result : Negative / Negative

Positive Sample

i. Sample 1

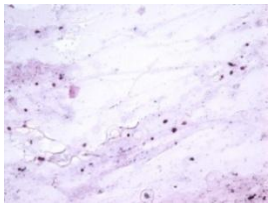
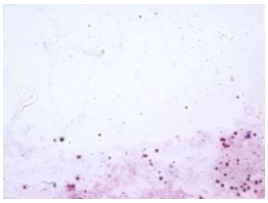
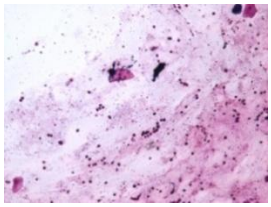
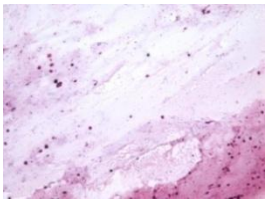





Image 1 Image 2 Image 3 Image 4

Pus cell

Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
<10 , score:0										
10-25, score:+1		X 24	X (24)							
>25, score:+2	X (25)			X 35	X (65)	X 121	X (38)	X 49	X (38)	X 57

Epithelial cell

Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
<10 , score:0	X (0)	X 1	X (0)	X 0	X (2)	X 3	X (0)	X 0	X (1)	X 1
10-25, score: -1										
>25, score: -2										

Result : Positive / Positive

ii. Sample 2

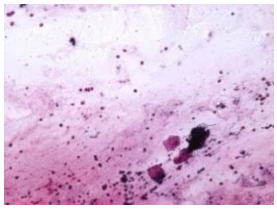
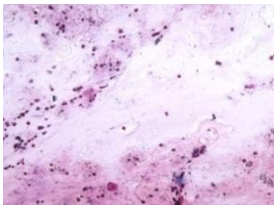
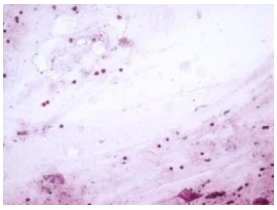
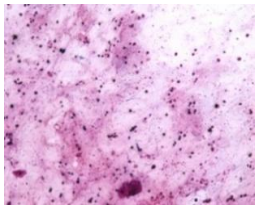





Image 1 Image 2 Image 3 Image 4

Pus cell

Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
<10 , score:0										
10-25, score:+1										
>25, score:+2	X (66)	X 95	X (79)	X 74	X (29)	X 38	X (109)	X 127	X (71)	X 84

Epithelial cell

Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
<10 , score:0	X (4)	X 3	X (0)	X 1	X (1)	X 2	X (1)	X 3	X (2)	X 2
10-25, score: -1										
>25, score: -2										

Result : Positive / Positive

iii. Sample 3

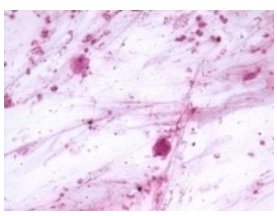
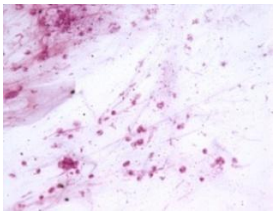
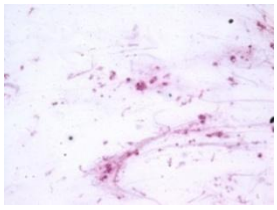
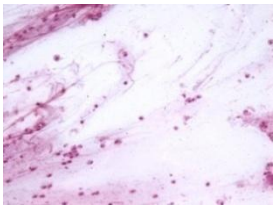





Image 1 Image 2 Image 3 Image 4

Pus cell

Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
<10 , score:0										
10-25, score:+1					X (12)					
>25, score:+2	X (25)	X 31	X (29)	X 44		X 28	X (48)	X 62	X (29)	X 41

Epithelial cell

Range/Score	Image 1		Image 2		Image 3		Image 4		Average (score)	
	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM	Sys	HUSM
<10 , score:0	X (1)	X 2	X (0)	X 1	X (0)	X 0	X (0)	X 0	X (0)	X 1
10-25, score: -1										
>25, score: -2										

Result : Positive / Positive

Validate by:

Rosliza Abd Rahman

Medical Laboratory Technologist U38

Appendix B

Coding for Pus Cell

```
ori_img1 =imread('C:\Documents and Settings\USER\My
Documents\MATLAB\SPUTUM_envx\POSITIVE\SAMPLE 1\1.jpg');
imshow(ori_img1), title('original img');
ori_img = imresize(ori_img1,0.2);
figure(1),imshow(ori_img1), title('ori_image');
figure(2),subplot(2,2,1),imshow(ori_img), title('smaller ori_image');

%color thresholding

handles.im1 = ori_img;
handles.im2 = ori_img;
red = handles.im2(:,:,1);
green = handles.im2(:,:,2);
blue = handles.im2(:,:,3);
handles.image_gray = rgb2gray(handles.im2);

[r c] = size(handles.image_gray);
for loop1 = 1:r
    for loop2 = 1:c
        if red(loop1,loop2) > 255||red(loop1,loop2)<0|| ...
           green(loop1,loop2) > 255||green(loop1,loop2)<100|| ...
           blue(loop1,loop2) > 255||blue(loop1,loop2)<100
           handles.im2(loop1,loop2,:)=0;
        end
    end
end

figure(2),subplot(2,2,2), imshow(handles.im2), title('color threshold image');

%substract image
handles.deltaImage = imsubtract(handles.im1,handles.im2);
figure(2),subplot(2,2,3), imshow(handles.deltaImage), title('image substract');

%image filtering
B = fspecial('disk',3);
blur_img = imfilter(handles.deltaImage,B,'replicate');
figure(2),subplot(2,2,4), imshow(blur_img), title('blur image');

%Convert Image from RGB Color Space to L*a*b* Color Space
cform = makecform('srgb2lab');
handles.lab_blur_img = applycform(blur_img,cform);

%Classify the Colors in 'a*b*' Space Using K-Means Clustering
ab = double(handles.lab_blur_img(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);
nColors = 2;
[cluster_idx cluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean','emptyaction','drop','start','unifor
m','Replicates',8);

%Label Every Pixel in the Image Using the Results from KMEANS
pixel_labels = reshape(cluster_idx,nrows,ncols);
```

```
%Create Images that Segment the original Image by Color
segmented_images = cell(1,3);
rgb_label = repmat(pixel_labels,[1 1 3]);

for k = 1:nColors
    color = ori_img;
    color(rgb_label ~= k) = 0;
    segmented_images{k} = color;
end

figure(3), subplot(2,1,1), (imshow(segmented_images{1})), title('objects in
cluster 1');
figure(3), subplot(2,1,2), (imshow(segmented_images{2})), title('objects in
cluster 2');

hold on;

choice = questdlg('Choose cluster that consists pus cells', ...
    'Choose cluster', ...
    'cluster1','cluster2','cluster2');
% Handle response
switch choice
    case 'cluster1'
        h=1

    case 'cluster2'
        h=2


end
figure(4),subplot(2,2,1), imshow(segmented_images{h}), title('proceed image');
handles.segmented_images{h} = segmented_images{h};
handles.gray = rgb2gray(handles.segmented_images{h});
figure(4),subplot(2,2,2),imshow(handles.gray), title('gray_img');

%gray2binary
handles.binaryImage = im2bw(handles.gray,0.1);
handles.binaryImage = imfill(handles.binaryImage, 'holes');
figure(4),subplot(2,2,3), imshow(handles.binaryImage), title('binary_img');

handles.BinaryImg2 = bwareaopen(handles.binaryImage, 20);
figure(4),subplot(2,2,4), imshow(handles.BinaryImg2),
title('nonoisebinary_img');

labeledImage = bwlabel(handles.BinaryImg2, 8);
figure(5),subplot(2,2,1), imshow(labeledImage, []), title('labelimg');
objMeasurements = regionprops(labeledImage, handles.gray, 'all');
numberOfobj = size(objMeasurements, 1);
hold on;
boundaries = bwboundaries(handles.BinaryImg2);
numberOfBoundaries = size(boundaries);
for k = 1 : numberOfBoundaries
    thisBoundary = boundaries{k};
    plot(thisBoundary(:,2), thisBoundary(:,1), 'g', 'LineWidth', 2);
end
hold off;

fontSize = 10;  % Used to control size of "obj number" labels put atop the
image.
labelShiftX = -7;   % Used to align the labels in the centers of the coins.
objECD = zeros(1, numberOfobj);
```

```
fprintf(1,'Obj #      Mean Intensity  Area   Perimeter    Centroid
Diameter\n');

for k = 1 : numberOfobj           % Loop through all blobs.
    % Find the mean of each blob.  (R2008a has a better way where you can pass
the original image
    % directly into regionprops.  The way below works for all versions
including earlier versions.)
    thisobjPixels = objMeasurements(k).PixelIdxList;  % Get list of pixels in
current blob.
    meanGL = mean(gray(thisobjPixels)); % Find mean intensity (in original
image!)
    meanGL2008a = objMeasurements(k).MeanIntensity; % Mean again, but only for
version >= R2008a

    objArea = objMeasurements(k).Area;      % Get area.
    objPerimeter = objMeasurements(k).Perimeter;        % Get perimeter.
    objCentroid = objMeasurements(k).Centroid;      % Get centroid.
    objECD(k) = sqrt(4 * objArea / pi);                 % Compute ECD -
Equivalent Circular Diameter.
    fprintf(1,'#%2d %17.1f %11.1f %8.1f %8.1f %8.1f % 8.1f\n', k, meanGL,
objArea, objPerimeter, objCentroid, objECD(k));
    % Put the "blob number" labels on the "boundaries" grayscale image.
    text(objCentroid(1) + labelShiftX, objCentroid(2), num2str(k), 'FontSize',
fontSize, 'FontWeight', 'Bold','Color','y');
end



%single cell
handles.labeledImage = bwlabel(handles.BinaryImg2, 8);
handles.objMeasurements = regionprops(handles.labeledImage, handles.gray,
'all');
handles.numberOfobj = size(handles.objMeasurements, 1);

handles.allobjIntensities = [handles.objMeasurements.MeanIntensity];
handles.allobjAreas = [handles.objMeasurements.Area];
handles.allobjPerimeters = [handles.objMeasurements.Perimeter];


handles.allowableIntensityIndexes = (handles.allobjIntensities > 90) &
(handles.allobjIntensities < 142);
handles.allowableAreaIndexes = (handles.allobjAreas > 20) &
(handles.allobjAreas < 100);
handles.allowablePerimeterIndexes = (handles.allobjPerimeters > 15) &
(handles.allobjPerimeters < 38);
handles.keeperIndexes = find(handles.allowableIntensityIndexes &
handles.allowableAreaIndexes & handles.allowablePerimeterIndexes);

handles.keeperobjImage = ismember(handles.labeledImage, handles.keeperIndexes);
handles.labeledPusImage = bwlabel(handles.keeperobjImage, 8);

handles.final = im2bw(handles.labeledPusImage,0.1);
figure(5),subplot(2,2,2), imshow(handles.final), title('pus cells img');

%count single cell
handles.B = bwboundaries(handles.final,'holes');
se = strel('disk',3);
handles.Ifill = handles.final;
closeBW = imclose(handles.Ifill,se);
t = length(handles.B)

figure(5),subplot(2,2,3),imshow(handles.final);
```

```
%detect boundaries of object
B4 = bwboundaries(handles.final,'holes');
se = strel('disk',3);
handles.Ifill4 = handles.final;
handles.closeBW4 = imclose(handles.Ifill4,se);

handles.BWoutline = bwperim(handles.closeBW4);
%figure(5),subplot(2,2,4),imshow(handles.BWoutline),
Segout = ori_img;
Segout(handles.BWoutline)= 0;

figure(6),imshow(Segout), title('outlined original image');
total = t


prompt2 = {['NUMBER OF PUS CELLS DETECTED = ' int2str(total)],'',...
    'Please click on Add Image/Results button'};
dlg_title2 = ('Results');
msgbox(prompt2,dlg_title2);
```

Appendix C

GUI coding for complete Sputum Quality Testing System

```
function varargout = system(varargin)

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @system_OpeningFcn, ...
                   'gui_OutputFcn',  @system_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before system is made visible.
function system_OpeningFcn(hObject, eventdata, handles, varargin)

axes(handles.axes10); imshow('C:\Documents and Settings\USER\My
Documents\Automated Sputum Quality Testing\ump.jpg');
axes(handles.axes11); imshow('C:\Documents and Settings\USER\My
Documents\Automated Sputum Quality Testing\visis.jpg');
axes(handles.axes12); imshow('C:\Documents and Settings\USER\My
Documents\Automated Sputum Quality Testing\husm.jpg');
axes(handles.axes13); imshow('C:\Documents and Settings\USER\My
Documents\Automated Sputum Quality Testing\logo.jpg');
set(handles.axes1,'Visible','off');
set(handles.axes2,'Visible','off');
set(handles.axes14,'Visible','off');
set(handles.axes4,'Visible','off');
set(handles.axes6,'Visible','off');
set(handles.axes7,'Visible','off');
set(handles.axes8,'Visible','off');
set(handles.axes9,'Visible','off');
set(handles.text1,'visible','off');
set(handles.text2,'visible','off');
set(handles.text3,'visible','off');
set(handles.text4,'visible','off');
set(handles.load,'enable','off');
set(handles.run,'enable','off');
set(handles.add,'enable','off');
handles.output = hObject;
```

```
% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = system_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;


% --- Executes on button press in start.
function start_Callback(hObject, eventdata, handles)
set(handles.text5,'visible','off');
set(handles.text32,'visible','on');
set(handles.load,'enable','on');
k = 5;
handles.k = k;
set(handles.load,'string','Load Image1');
prompt = {'4 sample of sputum image will be loaded one by
one.','','Please click on Load Image1 button to load first image.'};
dlg_title = ('Instruction');
msgbox(prompt,dlg_title);
guidata(hObject, handles);

% --- Executes on button press in load.
function load_Callback(hObject, eventdata, handles)

set(handles.start,'enable','off');
[FileName,PathName] = uigetfile({'*.jpg'},'Load Image File');

if (FileName==0) % cancel pressed
    return;
end

handles.fullPath = [PathName FileName];
ori_img1 = imread(handles.fullPath);
handles.ori_img1 = ori_img1;
handles.ori_img = imresize(handles.ori_img1,0.2);
axes(handles.axes1); imshow(handles.ori_img), title('original image');
set(handles.text1,'visible','off');
set(handles.run,'enable','on');
guidata(hObject, handles);


% --- Executes on button press in run.
function run_Callback(hObject, eventdata, handles)

wait=0;
set(handles.text1,'visible','off');
set(handles.text2,'visible','off');
set(handles.load,'enable','off');

handles.deltaImage = handles.ori_img;
handles.gray_delt = rgb2gray(handles.deltaImage);
```

```
handles.binaryImagee_delt = handles.gray_delt < 140;
handles.BinaryImagee_delt = imfill(handles.binaryImagee_delt, 'holes');
handles.Binary3_delt = bwareaopen(handles.BinaryImagee_delt,25);

handles.B = bwboundaries(handles.Binary3_delt,'holes');
handles.numberOfobj = length(handles.B);
handles.numberOfobj

handles.by = bwareaopen(handles.Binary3_delt,60);
handles.C = bwboundaries(handles.by,'holes');
handles.numberOfobj2 = length(handles.C);
handles.numberOfobj2
%obj3=pus
handles.obj3 = handles.numberOfobj - handles.numberOfobj2;
handles.obj3

if handles.obj3 >= handles.numberOfobj2
    method=2;
else
    method=1;
end

method
handles.deltaImage = handles.ori_img;
handles.gray = rgb2gray(handles.deltaImage);
handles.grays = handles.gray;
axes(handles.axes2); imshow(handles.gray), title('gray image');

%start with method
if method == 1
thresholdValue = 200;
else
  thresholdValue = 100;
end

binaryImage2 = handles.gray < thresholdValue;
handles.binaryImage2 = imfill(binaryImage2, 'holes');
handles.BinaryImagee = handles.binaryImage2 ;
axes(handles.axes2); imshow(handles.BinaryImagee), title('binary
image');

handles.labeledImaget = bwlabel(handles.binaryImage2, 8);
handles.objMeasurementst = regionprops(handles.labeledImaget,
handles.gray, 'all');
allobjAreast = [handles.objMeasurementst.Area];
handles.allowableAreaIndexest = (allobjAreast > 330);
handles.keeperIndexest = find(handles.allowableAreaIndexest);
handles.keeperobjImaget = ismember(handles.labeledImaget,
handles.keeperIndexest);
handles.labeledPusImaget = bwlabel(handles.keeperobjImaget, 8);
axes(handles.axes2); imshow(handles.labeledPusImaget), title('label
image');

handles.fg = im2bw(handles.labeledPusImaget);
[hg, map] = gray2ind(handles.fg,65534);
```

```
handles.RGBl = ind2rgb(hg,map);
handles.RGBk = im2uint8(handles.RGBl);

%color threshold:swap background and object
 handles.RGB2 = handles.RGBk;
 handles.RGB = handles.RGBk;
 red = handles.RGB(:,:,1);
 green = handles.RGB(:,:,2);
 blue = handles.RGB(:,:,3);
 handles.image_gray = rgb2gray(handles.RGB);

 [r c] = size(handles.image_gray);
  for loop1 = 1:r
  for loop2 = 1:c
       if red(loop1,loop2) > 256||red(loop1,loop2)<230|| ...
          green(loop1,loop2) > 256||green(loop1,loop2)<230|| ...
          blue(loop1,loop2) > 256||blue(loop1,loop2)<230
          handles.RGB(loop1,loop2,:)=255;
        end
    end
  end

axes(handles.axes2); imshow(handles.RGB), title('swap image');

handles.RGB = handles.RGB - handles.RGB2;
axes(handles.axes2); imshow(handles.RGB), title('scn image');
handles.ori_img5 = handles.ori_img - handles.RGB;

red = handles.RGB(:,:,1);
green = handles.RGB(:,:,2);
blue = handles.RGB(:,:,3);
image_gray = rgb2gray(handles.RGB);

handles.oris_img5 = handles.ori_img5;
handles.gray = rgb2gray(handles.oris_img5);
handles.bws = im2bw(handles.gray,0.1);
handles.bw = imfill(handles.bws,'holes');
axes(handles.axes2), imshow(handles.bw), title('ori image');

handles.BinaryImg2 = handles.bw;
handles.labeledImage = bwlabel(handles.BinaryImg2, 8);
handles.objMeasurements = regionprops(handles.labeledImage,
handles.gray, 'all');
handles.numberOfobj = size(handles.objMeasurements, 1);

handles.allobjIntensities = [handles.objMeasurements.MeanIntensity];
handles.allobjIntensities1 = [handles.objMeasurements.MeanIntensity];
handles.allobjAreas = [handles.objMeasurements.Area];
handles.allowableIntensityIndexes = (handles.allobjIntensities > 0) &
(handles.allobjIntensities < 255);
handles.allowableIntensityIndexes1 = (handles.allobjIntensities > 0)
&(handles.allobjIntensities1 <140);
handles.allowableIntensityIndexes11 = (handles.allobjIntensities > 160)
&(handles.allobjIntensities1 <170);
```

```
handles.allowableAreaIndexes = (handles.allobjAreas > 4000) &
(handles.allobjAreas < 30000);

handles.keeperIndexes2 = find(handles.allowableIntensityIndexes);
handles.keeperobjImage2 = ismember(handles.labeledImage,
handles.keeperIndexes2);
handles.labeledPusImage2 = bwlabel(handles.keeperobjImage2, 8);
axes(handles.axes2),imshow(handles.labeledPusImage2, []);
handles.final2 = im2bw(handles.labeledPusImage2,0.1);
axes(handles.axes2),imshow(handles.final2), title('bw img');

%for single cell
handles.objMeasurements3 = regionprops(handles.final2, 'area');
handles.numberOfobj3 = size(handles.objMeasurements3, 1);
handles.allobjAreas4 = [handles.objMeasurements3.Area];
handles.allowableAreaIndexes4 = (handles.allobjAreas4 > 355) &
(handles.allobjAreas4 < 1300);
handles.keeperIndexes4 = find(handles.allowableAreaIndexes4);
handles.keeperobjImage4 = ismember(handles.labeledPusImage2,
handles.keeperIndexes4);
handles.labeledPusImage4 = bwlabel(handles.keeperobjImage4, 8);
handles.final4 = im2bw(handles.labeledPusImage4,0.1);
axes(handles.axes2),imshow(handles.final4), title('single pus cells
img');

%count single cell
handles.B = bwboundaries(handles.final4,'holes');
t = length(handles.B);
t

%for double cell
handles.objMeasurements4 = regionprops(handles.final2, 'area');
handles.numberOfobj4 = size(handles.objMeasurements4, 1);
handles.allobjAreas5 = [handles.objMeasurements4.Area];
handles.allowableAreaIndexes5 = (handles.allobjAreas5 > 1300) &
(handles.allobjAreas5 < 3000);
handles.keeperIndexes5 = find(handles.allowableAreaIndexes5);
handles.keeperobjImage5 = ismember(handles.labeledPusImage2,
handles.keeperIndexes5);
handles.labeledPusImage5 = bwlabel(handles.keeperobjImage5, 8);
handles.final5 = im2bw(handles.labeledPusImage5,0.1);
axes(handles.axes2),imshow(handles.final5), title('double pus cells
img');

%count double cell
handles.C = bwboundaries(handles.final5,'holes');
t2 = 2*length(handles.C);
t2

%for triple cell
handles.objMeasurements5 = regionprops(handles.final2, 'area');
handles.numberOfobj5 = size(handles.objMeasurements5, 1);
handles.allobjAreas6 = [handles.objMeasurements5.Area];
handles.allowableAreaIndexes6 = (handles.allobjAreas6 > 3100) &
(handles.allobjAreas6 < 5000);
handles.keeperIndexes6 = find(handles.allowableAreaIndexes6);
```

```
handles.keeperobjImage6 = ismember(handles.labeledPusImage2,
handles.keeperIndexes6);
handles.labeledPusImage6 = bwlabel(handles.keeperobjImage6, 8);
handles.final6 = im2bw(handles.labeledPusImage6,0.1);
axes(handles.axes2),imshow(handles.final6), title('triple pus cells
img');

%count triple cell
handles.D = bwboundaries(handles.final6,'holes');
t3 = 3*length(handles.D);
t3

totalepi = t + t2 + t3;
totalepi

if totalepi == 0
    totalepi=totalepi;
    numberOfobj1=0;

else

for u=1 : handles.numberOfobj;
   handles.keeperIndexes1 = find(handles.allowableAreaIndexes &
handles.allowableIntensityIndexes1);
   handles.keeperIndexes11 = find(handles.allowableAreaIndexes &
handles.allowableIntensityIndexes11);
   handles.keeperobjImage1 = ismember(handles.labeledImage,
handles.keeperIndexes1);
   handles.keeperobjImage11 = ismember(handles.labeledImage,
handles.keeperIndexes11);
   handles.labeledPusImage1 = bwlabel(handles.keeperobjImage1, 8);
   handles.labeledPusImage11 = bwlabel(handles.keeperobjImage11, 8);
   axes(handles.axes2), imshow(handles.labeledPusImage11, []), title('n
pus cells img');
   handles.final0 = im2bw(handles.labeledPusImage1,0.1);
   handles.final11 = im2bw(handles.labeledPusImage11,0.1);

end


handles.final = imadd(handles.final0,handles.final11);
axes(handles.axes2), imshow(handles.final),
handles.objMeasurements1 = regionprops(handles.final, 'area');
numberOfobj1 = size(handles.objMeasurements1, 1);
numberOfobj1

end

if numberOfobj1==0
    totalepi2 = totalepi;
```

```
else

   for k=1 : numberOfobj1;
    handles.objArea(k) = handles.objMeasurements1(k).Area;

   end

    j = numberOfobj1;
    k = [1:j];
    handles.s = sum(handles.objArea(k));
    handles.sk = int2str(handles.s/1100);
    t4 = str2double(handles.sk);
    t4
    totalepi2 = t4 + totalepi;


end


totalepi2
handles.totalepi2 = totalepi2;

handles.qye =totalepi2;
handles.qyie =totalepi2;
handles.qyi1e =totalepi2;
handles.qyi2e =totalepi2;


%pus cell
handles.thresholdValue = 200;
handles.binaryImage = handles.grays < handles.thresholdValue;
handles.BinaryImagee2 = handles.binaryImage;

handles.BinaryImagee = imfill(handles.BinaryImagee2, 'holes');
axes(handles.axes2),imshow(handles.BinaryImagee), title('binary_img');

handles.labeledImagee = bwlabel(handles.BinaryImagee, 8);
handles.objMeasurementst = regionprops(handles.labeledImagee,
handles.gray, 'all');

handles.allobjAreast = [handles.objMeasurementst.Area];
if method == 1
    handles.allowableAreaIndexest = (handles.allobjAreast > 330);
else
    handles.allowableAreaIndexest = (handles.allobjAreast > 800) &
(handles.allobjAreast < 810);
end

handles.keeperIndexest = find(handles.allowableAreaIndexest);
handles.keeperobjImagee = ismember(handles.labeledImagee,
handles.keeperIndexest);
handles.labeledPusImagee = bwlabel(handles.keeperobjImagee, 8);
axes(handles.axes2), imshow(handles.labeledPusImagee, []), title('pus
cells img');
```

```
handles.fg = im2bw(handles.labeledPusImagee);

[hg, map] = gray2ind(handles.fg,65534);
handles.RGBk = ind2rgb(hg,map);
handles.RGBk = im2uint8(handles.RGBk);
axes(handles.axes2), imshow(handles.RGBk), title('RGBk img');

handles.fin = imadd(handles.ori_img,handles.RGBk);
axes(handles.axes2), imshow(handles.fin), title('add img');

%color thresholding

handles.RGB2 = handles.fin;
handles.RGB = handles.fin;
red = handles.RGB(:,:,1);
green = handles.RGB(:,:,2);
blue = handles.RGB(:,:,3);
handles.image_gray = rgb2gray(handles.RGB);

[r c] = size(handles.image_gray);
for loop1 = 1:r
    for loop2 = 1:c
        if red(loop1,loop2) > 255||red(loop1,loop2)<0|| ...
           green(loop1,loop2) > 255||green(loop1,loop2)<100|| ...
           blue(loop1,loop2) > 255||blue(loop1,loop2)<100
           handles.RGB(loop1,loop2,:)=0;
        end
    end
end

axes(handles.axes2),imshow(handles.RGB), title('color threshold
image');

%substract image
handles.im1 = handles.RGB2;
handles.im2 = handles.RGB;
handles.deltaImage = imsubtract(handles.im1,handles.im2);
axes(handles.axes2), imshow(handles.deltaImage), title('image
substract');

handles.adj_img = handles.deltaImage;
B = fspecial('disk',3);
blur_img = imfilter(handles.adj_img,B,'replicate');
axes(handles.axes2), imshow(blur_img), title('blur image');

%Convert Image from RGB Color Space to L*a*b* Color Space
cform = makecform('srgb2lab');
handles.blur_img = blur_img;
handles.cform = cform;
lab_blur_img = applycform(handles.blur_img,handles.cform);

%Classify the Colors in 'a*b*' Space Using K-Means Clustering
handles.lab_blur_img = lab_blur_img;
ab = double(handles.lab_blur_img(:,:,2:3));
handles.ab = ab;
```

```
nrows = size(handles.ab,1);
ncols = size(handles.ab,2);
handles.nrows = nrows; handles.ncols = ncols;
ab = reshape(handles.ab,handles.nrows*handles.ncols,2);
nColors = 2;
handles.ab = ab;
handles.nColors = nColors;
[cluster_idx cluster_center] =
kmeans(handles.ab,handles.nColors,'distance','sqEuclidean','emptyaction
','drop','start','uniform','Replicates',8);
%number of cluster=2

%Label Every Pixel in the Image Using the Results from KMEANS
handles.cluster_idx = cluster_idx;
handles.nrows = nrows; handles.ncols = ncols;
pixel_labels =
reshape(handles.cluster_idx,handles.nrows,handles.ncols);

%Create Images that Segment the original Image by Color
segmented_images = cell(1,3);
handles.pixel_labels = pixel_labels;
rgb_label = repmat(handles.pixel_labels,[1 1 3]);
handles.rgb_label = rgb_label;
for k = 1:nColors
    color = handles.ori_img;
    color(handles.rgb_label ~= k) = 0;
    segmented_images{k} = color;
end

axes(handles.axes2), (imshow(segmented_images{1})), title('objects in
cluster 1');
axes(handles.axes14), (imshow(segmented_images{2})), title('objects in
cluster 2');

set(handles.text35,'visible','off');
set(handles.text17,'visible','off');
set(handles.text18,'visible','off');
set(handles.text19,'visible','off');
set(handles.text20,'visible','off');
set(handles.text3,'visible','off');

hold on;

choice = questdlg('Choose cluster that consists pus cells.', ...
    'Choose cluster', ...
    'cluster1','cluster2','cancel','cancel');
% Handle response
switch choice
    case 'cluster1'
        h=1

    case 'cluster2'
        h=2

    case 'cancel'
```

```
        cla(handles.axes1),
        set(handles.axes1,'Visible','off');
        set(handles.text1,'visible','on');
        cla(handles.axes2);
        set(handles.axes2,'Visible','off');
        set(handles.text2,'visible','on');
        cla(handles.axes14);
        set(handles.axes14,'Visible','off');
        set(handles.text3,'visible','on');
        guidata(hObject, handles);
        msgbox('Please reload the image.','warn');
        set(handles.load,'enable','on');
        %clear;
        wait = 1;


end


if wait ==1
    'error'
else
axes(handles.axes2), imshow(segmented_images{h}), title('proceed
image');
cla(handles.axes14);
set(handles.text3,'visible','on');
set(handles.text17,'visible','on');
set(handles.text18,'visible','on');
set(handles.text19,'visible','on');
set(handles.text20,'visible','on');
handles.segmented_images{h} = segmented_images{h};
handles.gray = rgb2gray(handles.segmented_images{h});
handles.gray3 = handles.gray;
handles.gray2 = handles.gray;
handles.handles.grayn = handles.gray;
axes(handles.axes2), imshow(handles.gray), title('gray_img');

%thresholdValue = 90;
handles.binaryImage = im2bw(handles.gray,0.1);
handles.binaryImage = imfill(handles.binaryImage, 'holes');
axes(handles.axes2), imshow(handles.binaryImage), title('binary_img');

handles.BinaryImg2 = bwareaopen(handles.binaryImage, 20);
%start with single cell
handles.labeledImage = bwlabel(handles.BinaryImg2, 8);
handles.labeledImage3 = handles.labeledImage;
handles.labeledImage2 = handles.labeledImage;
handles.labeledImagen = handles.labeledImage;
handles.labeledImagen = handles.labeledImage;


handles.objMeasurements = regionprops(handles.labeledImage,
handles.gray, 'all');
handles.objMeasurements3 = regionprops(handles.labeledImage3,
handles.gray3, 'all');
```

```
handles.objMeasurements2 = regionprops(handles.labeledImage2,
handles.gray2, 'all');
handles.objMeasurementsn = regionprops(handles.labeledImagen,
handles.handles.grayn, 'all');
handles.numberOfobj = size(handles.objMeasurements, 1);

handles.allobjIntensities = [handles.objMeasurements.MeanIntensity];
handles.allobjAreas = [handles.objMeasurements.Area];
handles.allobjPerimeters = [handles.objMeasurements.Perimeter];
if method == 1
handles.allowableIntensityIndexes = (handles.allobjIntensities > 83) &
(handles.allobjIntensities < 107);
handles.allowableIntensityIndexes2 = (handles.allobjIntensities > 109)
& (handles.allobjIntensities < 172);
handles.allowableAreaIndexes = (handles.allobjAreas > 13) &
(handles.allobjAreas < 121);
handles.allowablePerimeterIndexes = (handles.allobjPerimeters > 10) &
(handles.allobjPerimeters < 39);
handles.keeperIndexes = find(handles.allowableIntensityIndexes &
handles.allowableAreaIndexes & handles.allowablePerimeterIndexes);
handles.keeperIndexes2 = find(handles.allowableIntensityIndexes2 &
handles.allowableAreaIndexes & handles.allowablePerimeterIndexes);

handles.keeperobjImage = ismember(handles.labeledImage,
handles.keeperIndexes);
handles.keeperobjImage2 = ismember(handles.labeledImage,
handles.keeperIndexes2);
handles.labeledPusImage1 = bwlabel(handles.keeperobjImage, 8);
handles.labeledPusImage2 = bwlabel(handles.keeperobjImage2, 8);
handles.labeledPusImage = handles.labeledPusImage1 +
handles.labeledPusImage2;
axes(handles.axes2), imshow(handles.labeledPusImage, []), title('pus
cells img');

else
handles.allowableIntensityIndexes = (handles.allobjIntensities > 90) &
(handles.allobjIntensities < 142);
handles.allowableAreaIndexes = (handles.allobjAreas > 20) &
(handles.allobjAreas < 100);
handles.allowablePerimeterIndexes = (handles.allobjPerimeters > 15) &
(handles.allobjPerimeters < 38);
handles.keeperIndexes = find(handles.allowableIntensityIndexes &
handles.allowableAreaIndexes & handles.allowablePerimeterIndexes);

handles.keeperobjImage = ismember(handles.labeledImage,
handles.keeperIndexes);
handles.labeledPusImage = bwlabel(handles.keeperobjImage, 8);
axes(handles.axes2), imshow(handles.labeledPusImage, []), title('pus
cells img');
end

handles.final = im2bw(handles.labeledPusImage,0.1);

%count single cell
handles.B = bwboundaries(handles.final,'holes');
se = strel('disk',3);
```

```
handles.Ifill = handles.final;
closeBW = imclose(handles.Ifill,se);
t = length(handles.B);
t

%start combine with double cell
handles.allobjIntensities2 = [handles.objMeasurements2.MeanIntensity];
handles.allobjAreas2 = [handles.objMeasurements2.Area];
handles.allobjPerimeters2 = [handles.objMeasurements2.Perimeter];

handles.allowableIntensityIndexes2 = (handles.allobjIntensities2 > 144)
& (handles.allobjIntensities2 < 150);
handles.allowableAreaIndexes2 = (handles.allobjAreas2 > 100) &
(handles.allobjAreas2 < 150);
handles.allowablePerimeterIndexes2 = (handles.allobjPerimeters2 > 28) &
(handles.allobjPerimeters2 < 60);
handles.keeperIndexes2 = find(handles.allowableIntensityIndexes2 &
handles.allowableAreaIndexes2 & handles.allowablePerimeterIndexes2);
handles.keeperobjImage2 = ismember(handles.labeledImage2,
handles.keeperIndexes2);
handles.labeledPusImage2 = bwlabel(handles.keeperobjImage2, 8);
axes(handles.axes2), imshow(handles.labeledPusImage2, []), title('pus
cells2 img');
handles.final2 = im2bw(handles.labeledPusImage2,0.1);

%count double cell
handles.B2 = bwboundaries(handles.final2,'holes');
se = strel('disk',3);
handles.Ifill2 = handles.final2;
closeBW2 = imclose(handles.Ifill2,se);
t2 = 2*length(handles.B2);
t2

handles.final4 = imadd(handles.final,handles.final2);
handles.final6 = handles.final4;

axes(handles.axes2),imshow(handles.final6);

%detect boundaries of object
B4 = bwboundaries(handles.final6,'holes');
se = strel('disk',3);
handles.Ifill4 = handles.final6;
handles.closeBW4 = imclose(handles.Ifill4,se);

handles.BWoutline = bwperim(handles.closeBW4);
axes(handles.axes2),imshow(handles.BWoutline),
Segout = handles.ori_img;
Segout(handles.BWoutline)= 0;

axes(handles.axes2),imshow(Segout), title('outlined original image');
total = t + t2;
total
handles.total = total;

cla(handles.axes1),
```

```
cla(handles.axes2),
set(handles.text1,'visible','off');
set(handles.text2,'visible','off');
set(handles.text3,'visible','off');
axes(handles.axes4), imshow(Segout);
set(handles.text4,'visible','on');

handles.qy = total;
handles.qyi = total;
handles.qyi1 = total;
handles.qyi2 = total;

set(handles.run,'enable','off');
set(handles.add,'enable','on');
prompt2 = {['NUMBER OF PUS CELLS DETECTED = ' int2str(total)],'',...
    ['NUMBER OF EPITHELIAL CELLS DETECTED = '
int2str(totalepi2)],'','Please click on Add Image/Results button'};
dlg_title2 = ('Results');
msgbox(prompt2,dlg_title2);

end
guidata(hObject, handles);



% --- Executes on button press in add.
function add_Callback(hObject, eventdata, handles)
cla(handles.axes1);
cla(handles.axes2);
cla(handles.axes14);
cla(handles.axes4);
set(handles.text1,'visible','on');
set(handles.text2,'visible','on');
set(handles.text3,'visible','on');
set(handles.text4,'visible','off');
set(handles.load,'enable','on');



handles.k = (handles.k)-1;
fu = handles.k;
if fu == 4
    %get(hObject,'String')
    qy1 = handles.qy;
    qy1e = handles.qye;
    L = eval( int2str(handles.total));
    LK = eval( int2str(handles.totalepi2));
    axes(handles.axes6), imshow(handles.ori_img);
    set(handles.text33,'visible','on');
    set(handles.text29,'visible','on');
    set(handles.text30,'visible','on');

    set(handles.text11,'visible','on');
    set(handles.text7,'visible','on');
    set(handles.text11,'string',L);%image1
```

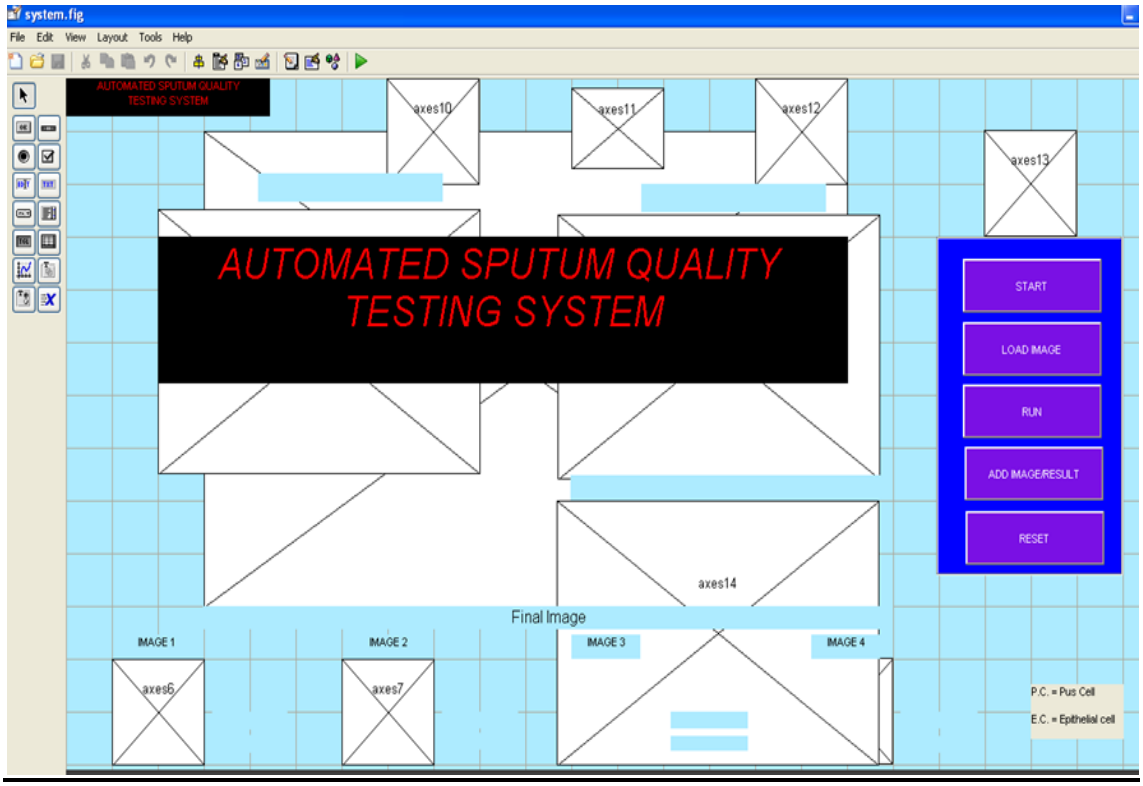
```
    set(handles.text7,'string','P.C. = ');
    set(handles.text12,'visible','on');
    set(handles.text13,'visible','on');
    set(handles.text13,'string',LK);%image1
    set(handles.text12,'string','E.C. = ');
    qy1
    qy1e
    handles.qy1 = qy1;
    handles.qy1e = qy1e;
    msgbox('Please load second image','','warn');
    set(handles.load,'string','Load Image2');
    set(handles.add,'enable','off');
    guidata(hObject, handles);



else

    if fu == 3
    L = eval( int2str(handles.total));
    LK = eval( int2str(handles.totalepi2));
    axes(handles.axes7), imshow(handles.ori_img);
    set(handles.text34,'visible','on');

    set(handles.text14,'visible','on');
    set(handles.text8,'visible','on');
    set(handles.text14,'string',L);%image1
    set(handles.text8,'string','P.C. = ');
    set(handles.text15,'visible','on');
    set(handles.text16,'visible','on');
    set(handles.text16,'string',LK);%image1
    set(handles.text15,'string','E.C. = ');
    qy2 = (handles.qy1) + (handles.qyi);
    qy2e = (handles.qy1e) + (handles.qyie);
    qy2
    qy2e
    handles.qy2 = qy2;
    handles.qy2e = qy2e;
    msgbox('Please load third image','','warn');
    set(handles.load,'string','Load Image3');
    set(handles.add,'enable','off');
    guidata(hObject, handles);


    else
        if fu == 2
        L = eval( int2str(handles.total));
        LK = eval( int2str(handles.totalepi2));
        axes(handles.axes8); imshow(handles.ori_img);
        set(handles.text35,'visible','on');

        set(handles.text17,'visible','on');
        set(handles.text20,'visible','on');
        set(handles.text17,'string',L);%image1
        set(handles.text20,'string','P.C. = ');
```

```
    set(handles.text18,'visible','on');
    set(handles.text19,'visible','on');
    set(handles.text19,'string',LK);%image1
    set(handles.text18,'string','E.C. = ');
    qy3 = (handles.qy2)+(handles.qyi1);
    qy3e = (handles.qy2e)+(handles.qyi1e);
    qy3
    qy3e
    handles.qy3 = qy3;
    handles.qy3e = qy3e;
    %handles.starts = 1;
    msgbox('Please load fourth image','','warn');
    set(handles.load,'string','Load Image4');
    set(handles.add,'enable','off');
    guidata(hObject, handles);

else

    if fu == 1
    set(handles.text35,'visible','on');
    L = eval( int2str(handles.total));
    LK = eval( int2str(handles.totalepi2));
    axes(handles.axes9), imshow(handles.ori_img);
    set(handles.text36,'visible','on');

    set(handles.text21,'visible','on');
    set(handles.text24,'visible','on');
    set(handles.text21,'string',L);%image1
    set(handles.text24,'string','P.C. = ');
    set(handles.text23,'visible','on');
    set(handles.text22,'visible','on');
    set(handles.text22,'string',LK);%image1
    set(handles.text23,'string','E.C. = ');
    qy4 =(handles.qy3) + (handles.qyi2);
    qy4e =(handles.qy3e) + (handles.qyi2e);
    qy4
    qy4e
    handles.qy4 = qy4;
    handles.qy4e = qy4e;
    set(handles.load,'string','Load Image');
    answer = handles.qy4/4;
    answere = handles.qy4e/4;
    guidata(hObject, handles);

    if answer >= 25
        scoreA = 2;


    else if answer >= 10
        scoreA = 1;

        else
          scoreA = 0
    end
    end
```

```
        if answere >= 25
            scoreC = -2;

        else if answere >= 10
                scoreC = -1;

            else
                scoreC = 0;

            end
        end

        prompte = {'Choose type of Macroscopy.','','Type A :Mucoid,
Mucopurulen, Purulent, or Blood Stain',...
            '', 'Type B :Watery'};
        choices = questdlg(prompte, ...
    'Choose Type', ...
    'Type A','Type B','Type B');

    switch choices
        case 'Type A'
        scoreB = 1;

        case 'Type B'
        scoreB = 0;

    end

    Totalscore = scoreA + scoreB + scoreC;
    Totalscore

    if Totalscore > 0

    prompt6 = {['NUMBER OF PUS CELLS(AVERAGE) = '
int2str(answer)],'',...
    ['NUMBER OF EPITHELIAL CELLS(AVERAGE) = '
int2str(answere)],'','',['Score A (Pus Cells) = '
int2str(scoreA)],'',...
    ['Score B (Macroscopy) = ' int2str(scoreB)],'',...
    ['Score C (Epithelial Cells) = ' int2str(scoreC)],'',['Total Score
= ' int2str(Totalscore)],'',...
    'Sample is POSITIVE.So,it can be cultured.'};
    dlg_title6 = ('Total Results');
    msgbox(prompt6,dlg_title6);

    else
    prompt8 = {['NUMBER OF PUS CELLS(AVERAGE) = '
int2str(answer)],'',...
    ['NUMBER OF EPITHELIAL CELLS(AVERAGE) = '
int2str(answere)],'','',['Score A (Pus Cells) = '
int2str(scoreA)],'',...
    ['Score B (Macroscopy) = ' int2str(scoreB)],'',...
```

```
    ['Score C (Epithelial Cells) = ' int2str(scoreC)],'',['Total Score
= ' int2str(Totalscore)],'',...
    'Sample is NEGATIVE.So,it cannot be cultured.'};
    dlg_title8 = ('Total Results');
    msgbox(prompt8,dlg_title8);

    end
        else
            'error'
     end
        end
    end



end



% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata, handles)
cla(handles.axes1),
set(handles.text33,'visible','off');
set(handles.text34,'visible','off');
set(handles.text35,'visible','off');
set(handles.text36,'visible','off');

set(handles.text32,'visible','off');
set(handles.axes1,'Visible','off');
set(handles.text1,'visible','on');
cla(handles.axes2);
set(handles.axes2,'Visible','off');
set(handles.text2,'visible','on');
cla(handles.axes14);
set(handles.axes14,'Visible','off');
set(handles.text3,'visible','on');
cla(handles.axes4);
set(handles.axes4,'Visible','off');
set(handles.text4,'visible','off');
set(handles.start,'enable','on');
set(handles.load,'enable','off');
set(handles.run,'enable','off');
set(handles.add,'enable','off');
set(handles.text5,'visible','on');
cla(handles.axes6),title('.');
cla(handles.axes7),title('.');
cla(handles.axes8),title('.');
cla(handles.axes9),title('.');
set(handles.text7,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
set(handles.text13,'visible','off');
set(handles.text7,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
```

```
set(handles.text13,'visible','off');
set(handles.text7,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
set(handles.text13,'visible','off');
set(handles.text7,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
set(handles.text13,'visible','off');
set(handles.text8,'visible','off');
set(handles.text14,'visible','off');
set(handles.text15,'visible','off');
set(handles.text16,'visible','off');
set(handles.text17,'visible','off');
set(handles.text18,'visible','off');
set(handles.text19,'visible','off');
set(handles.text20,'visible','off');
set(handles.text21,'visible','off');
set(handles.text22,'visible','off');
set(handles.text23,'visible','off');
set(handles.text24,'visible','off');
set(handles.text29,'visible','off');
set(handles.text30,'visible','off');
set(handles.text17,'string','');
set(handles.text18,'string','');
set(handles.text19,'string','');
set(handles.text20,'string','');

guidata(hObject, handles);
clear all
clc



function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```


system.fig
File Edit View Layout Tools Help
AUTOMATED SPUTUM QUALITY TESTING SYSTEM
axes10
axes11
axes12
axes13
AUTOMATED SPUTUM QUALITY TESTING SYSTEM
START
LOAD IMAGE
RUN
ADD IMAGE/RESULT
RESET
axes14
Final Image
IMAGE 1
IMAGE 2
IMAGE 3
IMAGE 4
axes6
axes7
P.C. = Pus Cell
E.C. = Epithelial cell

GUI Layout