AN IMAGE BASED SYSTEM TO OBJECTIVELY SCORE THE DEGREE OF SCALING IN PSORIASIS LESIONS

By

AZMI BIN MUHAMAD

FINAL PROJECT REPORT

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

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

AN IMAGE BASED SYSTEM TO OBJECTIVELY SCORE THE DEGREE OF SCALING IN PSORIASIS LESIONS

by

Azmi Bin Muhamad

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Mrs. Zazilah May
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2006

CERTIFICATION OF ORIGINALITY

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

Azmi Bin Muhamad

ABSTRACT

Skin diseases affect 20-30% of the population at any one time, seriously interfering with activities in 10%. Psoriasis is one of the most common skin diseases; it is non infectious, inflammatory disease of the skin. It is characterised by erythematous plaques with adherent silvery scales. There are known inherited genetic factors and several documented environmental factors that affect the clinical course of psoriasis. Men and women are equally affected and often seen between 15 to 15 years [1].The treatment of psoriasis requires the doctor to monitor the severity for each individual. They need to know the patient's condition either fully heals, moderate or bad.

A standard parameter for measuring the severity and coverage of psoriasis is used; namely Psoriasis Area and Severity Index (PASI). The PASI has been used for more than ten years and it is widely use around the world in the management of psoriasis. It is proved to be extremely effective in accessing psoriasis .Whilst it is not perfect, but it is the best method of assessment of psoriasis currently available [2]. This PASI have measure four variables which are the surface area of the psoriasis, the severity of Erythma (redness), the amount of scaling and the thickness of the plague. The surface area will be scored from 0 to 6 while the others will be scored from 0 to 4 depending on how much the percentage of psoriasis coverage and how severe the psoriasis is. Several problem are identified in current assessment of PASI, such problem are parameters measured are inaccurate, patient have to wait longer time for the result of further treatment and the evaluation in general performed on the base of clinical experience only.

This project report focuses on the assessment of scaliness using a computerized system for use in PASI calculation .The system used is image processing toolbox that available in MATLAB 7.0. The designed system will help dermatologist to objectively score the degree of scaling in psoriasis lesions. The project requires the author to import the digital image of infected skin, analyze the image in MATLAB 7.0, and provide an accurate score for PASI's scaling parameter. The methods used are the grayscale conversion, edge detection technique, image segmentation and pixels percentage calculation. The final results are the individual PASI score together with his personal data display on graphical user interface (GUI).

ACKNOWLEDGEMENT

First and foremost, I want to praise to the Al-Mighty Allah s.w.t for giving me the chance to finish my project. I would like to extend my gratitude to the Final Year Project (FYP) Committee for coming up a well structured FYP program for the final year students of UTP.

Most of my thanks obviously go out to my FYP supervisor, Mrs. Zazilah May from Electrical & Electronics Department for his continuous assistance, guidance and support throughout the entire duration of the project.

I would like to express my appreciation to my friend, Ahmad Tariq Abdul Mutalib for his assistance and encouragement. My housemate, Mohd Azri Mohd Kamil for helping me finishes the final project report. Thanks also to my technician, Miss Siti Hawa from Electrical & Electronics Department of UTP.

Finally, I would like to record my gratitude to my family members and friends for all their help and support throughout the duration of the project and also to the University Technology of PETRONAS that has given me the chance to study in engineering here.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background of Study

The skin can be infected by many type of diseases, no doubt if statistic showed that it is the most frequent reason for sick leave from work and is the most common industrial disease (Gawkrodger, 1997)[3].There are many types of skin diseases and one of the common skin diseases is psoriasis. An estimated 1–3% of the world's population has psoriasis at present. Often seen between 15 to 40 years [1]. There are known inherited genetic factors and several documented environmental factors that affect the clinical course of psoriasis. Men and women are equally affected. Siblings and offspring are at increased risk of developing psoriasis. Factors which may exacerbate psoriasis include human immunodeficiency virus (HIV) infection, physical trauma, infections including Streptococcus and Candida, certain drugs such as lithium, beta-blockers, antimalarials, and systemic corticosteroids.

The severity of psoriasis varies widely from one patient to another and over time. Severity is graded by the degree of skin involvement and the presence of joint involvement. A patient is considered to have severe psoriasis when more than 10% of the body surface is involved or when psoriatic arthritis is present. For the treatment of psoriasis, doctor need to monitor the severity for each individual They need to know the patient condition either fully heals, moderate or bad .So as a standard parameter for measuring this disease; they have use several technique of severity and coverage measurement. Such that techniques are The Psoriasis Area and Severity Index (PASI), Simplified Psoriasis Area and Severity Index (SPASI), National Psoriasis Foundation Psoriasis Score (NPF-PS), Physician static global assessment (PSGA) and Overall lesion assessment (OLA)[4].

The project implementation will be based on PASI score assessment only since it has been use for more than 10 years and is widely use around the world. Several problem are identified in the current methods, such problem are parameters measured are inaccurate, patient have to wait longer time for the result of further treatment and the evaluation in general performed on the base of clinical experience only.

So far, dermatologists have based the diagnosis of skin lesions on the visual assessment of pathological skin and the evaluation of macroscopic features. The clinical evaluation of the extent of the damages due to a skin in general performed on the base of clinical experience and using very basic and low-tech equipments for collecting objective measurements [5].An example of the common clinical approach to measure the *depth* of a skin lesion: a thin bar is introduced into the lesion until reaching the deepest point, selected according to clinician's evaluation; the bar is "marked" on the point corresponding to the potential skin surface height, and then measured by using a ruler [6]. The assessment of scaling parameter of infected skin also based on the observer's experience and on their visual acuity. All these procedures heavily rely on the skill of the clinicians, and therefore are not easily reproducible.

1.2 Problem Statement

The human vision lacks accuracy, reproducibility and quantification in gathering information from an image; thus systems that are able to evaluate images in an objective manner are obviously needed. MATLAB 7.0 is use in implementing the project. This project is beneficial in assisting medical experts in treating psoriasis patients. It can analyze the image of infected skin and give score as a basis of treatment and medication that is suitable.

1.3 Objectives

The objectives of this project are;

1) To build a system that are able to objectively score the degree of scaling in psoriasis lesions disease and calculate the Psoriasis Area and Severity Index (PASI) on patient;
2) To develop a cheap and practical technique of processing, comparing and analyzing skin disease to aid the clinical diagnosis;
3) To present an efficient method for the dermatological images characterization using MATLAB 7.0.

1.4 Scope of study

This study involves the design of MATLAB 7.0 algorithm capable to analyze and differentiate between textures of infected skin based on digital image. This texture analysis will be extracted and come out with a value that will be used in calculating the Psoriasis Area and Severity Index (PASI) score. The development of this technique will enable the monitoring of skin diseases in a telehealthcare application. As the beginning, the experiment process has been done base on data from the internet, which is digital image of infected skin. For the actual result, the input data must be from the medical specialist. This is done to ensure the system that the group will build is applicable in real situation.

CHAPTER 2
LITERATURE REVIEW/THEORY

2.1 Psoriasis

Psoriasis is a disease whose main symptom is gray or silvery flaky patches on the skin which are red inflamed underneath when scratched. Commonly affected areas include the scalp, elbows, knees, navel, and groin. Psoriasis is autoimmune in origin, and is not contagious [7]. Figure 2.1 below shows the most common affected areas of psoriasis diseases.

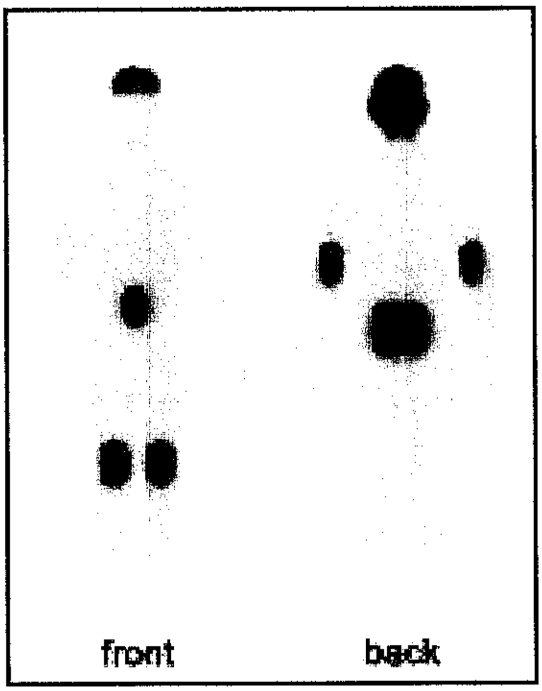


Figure 2.1: Areas typically affected by psoriasis

2.1.1 Types of Psoriasis

Plaque psoriasis

Plaque psoriasis is the most frequently diagnosed form of psoriasis. Plaque psoriasis can be recognized by the fact that the disease manifests itself as a very clearly defined area of skin which has a raised, red and inflamed section of skin covered in silvery scales. These scales are commonly referred to as *Plaques, Flakes or Scales.*

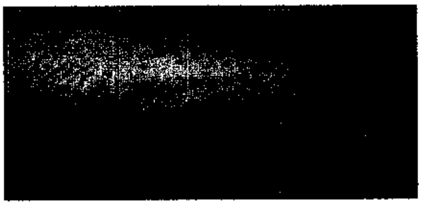


Figure 2.2: Image of Plague psoriasis

Guttate psoriasis

Guttate psoriasis is small, drop-like lesions appear on the trunk, limbs, and scalp. It is most often triggered by bacterial infections (for example, *Streptococcus*).

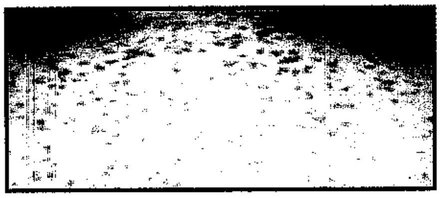


Figure 2.3: Image of Guttate psoriasis

Pustular psoriasis

For Pustular psoriasis, a blister of noninfectious pus appears on the skin. Attacks of Pustular psoriasis may be triggered by medications, infections, emotional stress, or exposure to certain chemicals. Pustular psoriasis may affect either small or large areas of the body.

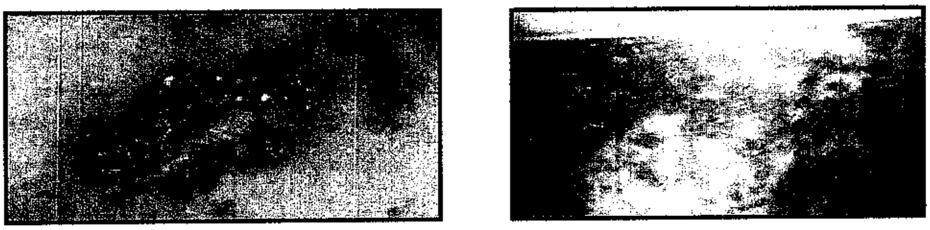


Figure 2.4: Image of Pustular psoriasis

Erythrodermic psoriasis

This type of psoriasis will appear as a widespread reddening and scaling of the skin, it is often accompanied by itching or pain. Erythrodermic psoriasis may be precipitated by severe sunburn, use of oral steroids (such as cortisone), or a drug-related rash.

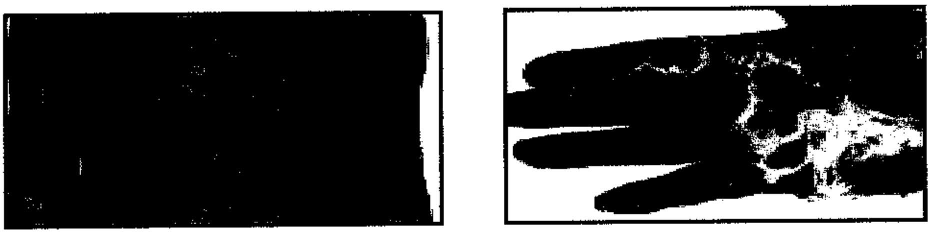


Figure 2.5: Image of Erythrodermic psoriasis

Inverse psoriasis

Its symptom appears as large, dry, smooth, vividly red plaques occur in the folds of the skin near the genitals, under the breasts, or in the armpits. Inverse psoriasis is related to increased sensitivity to friction and sweating and may be painful or itchy.

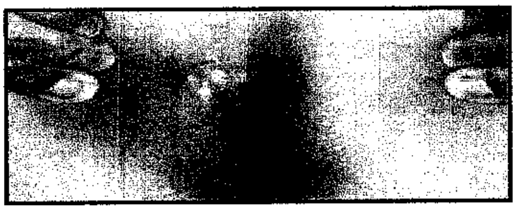


Figure 2.6: Image of Inverse psoriasis

2.1.2 Causes

Psoriasis is driven by the immune system, especially involving a type of white blood cell called a T cell. Normally, T cells help protect the body against infection and disease. In the case of psoriasis, T cells are put into action by mistake and become so active that they trigger other immune responses, which lead to inflammation and to rapid turnover of skin cells. These cells pile up on the surface of the skin, forming itchy patches or plaques. It is a non-contagious chronic inflammatory skin disease characterized by rapid growth of the outer skin layer (Figure 2.7), resulting in thick, silvery flakes of scale on raised pinkish red skin with well-defined margins (Figure 2.8). The first outbreak of psoriasis is often triggered by emotional or mental stress or physical skin injury, but heredity is a major factor as well. In about one-third of the cases, there is a family history of psoriasis. Researchers have studied a large number of families affected by psoriasis and identified genes linked to the disease. (Genes govern every bodily function and determine the inherited traits passed from parent to child.) People with psoriasis may notice that there are times when their skin worsens, then improves. Conditions that may cause flare-ups include infections, stress, and changes in climate that dry the skin. Also, certain medicines, including lithium and beta blockers, which are prescribed for high blood pressure, may trigger an outbreak or worsen the disease [8]. An activated immune system triggers the skin to reproduce every three to four days, it will build up on the outer layers (epidermis and keratin). The epidermis thickens, blood flow increases and reddens the skin, and silver-gray scales cover it (Figure 2.7).

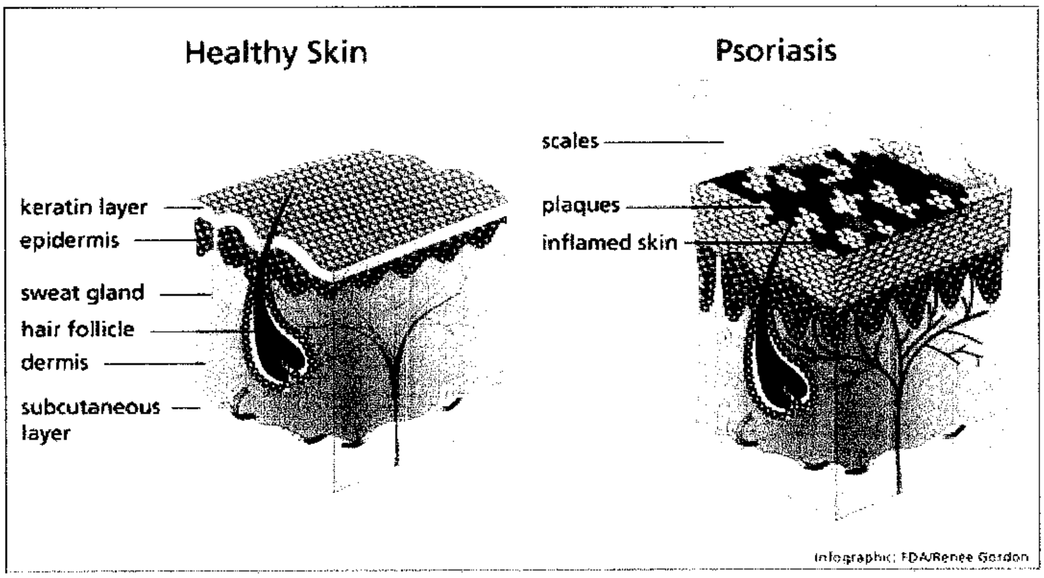


Figure 2.7: A healthy skin and the psoriasis infected skin

Figure 2.8: Chronic Plaque psoriasis

2.1.3 Treatment

Doctors generally treat psoriasis in steps based on the severity of the disease, size of the areas involved, type of psoriasis, and the patient's response to initial treatments. This is sometimes called the "1-2-3" approach. In step 1, medicines are applied to the skin (topical treatment). Step 2 uses light treatments (phototherapy). Step 3 involves taking medicines by mouth or injection that treat the whole immune system (called systemic therapy).

Over time, affected skin can become resistant to treatment, especially when topical corticosteroids are used. Also, a treatment that works very well in one person may have little effect in another. Thus, doctors often use a trial-and-error approach to find a treatment that works, and they may switch treatments periodically (for example, every 12 to 24 months) if a treatment does not work or if adverse reactions occur [8].

2.2 The Psoriasis Area and Severity Index (PASI)

For treatment procedure, doctor need to monitor the level of severity for each individual. They need to know the patient condition either fully heals, moderate or bad .So as a standard parameter for measuring this disease; they have use Psoriasis Area and Severity Index (PASI) as a technique to measure severity and coverage .The PASI would be calculated before, during, and after a treatment period in order to determine how well psoriasis responds to the treatment under test. In clinical trial, it is common that a "modified PASI" is use as the measurement and it is depend on the researcher.

For the PASI calculation, the body will be divided into four sections which are Legs (40% of person's skin), Trunk (30% of person's skin), Arms (20% of person's skin), and Head (10% of person's skin). Each of these areas is score by itself, and then the four score are combined into the final PASI [9]. The following figures show the PASI body regions.

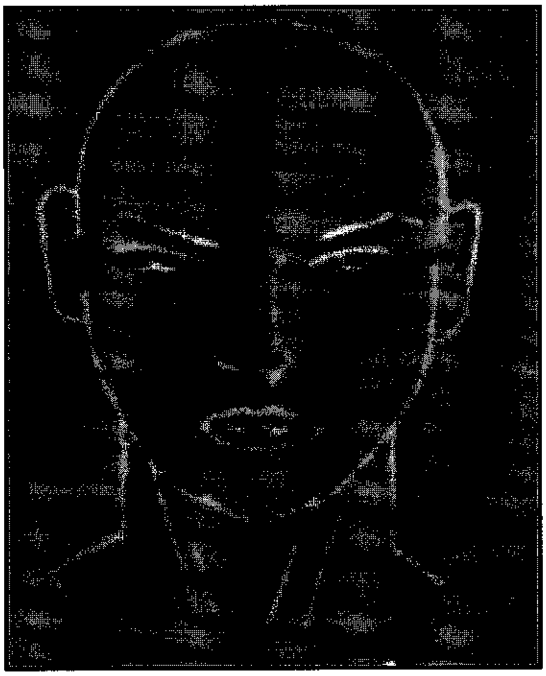


Figure 2.9: Head region

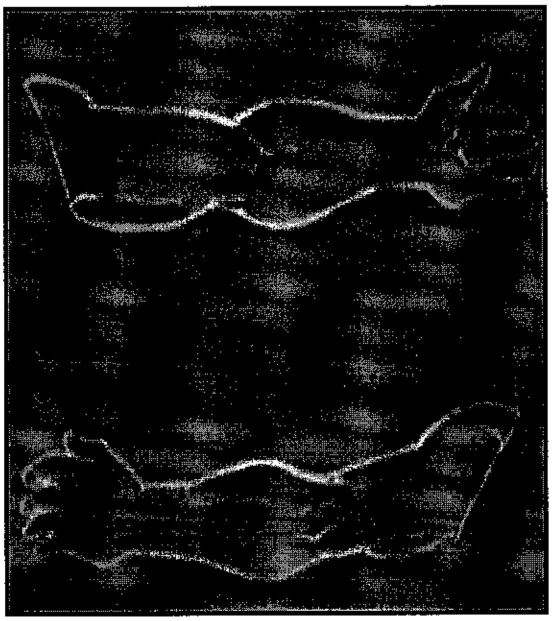


Figure 2.10: Arms region

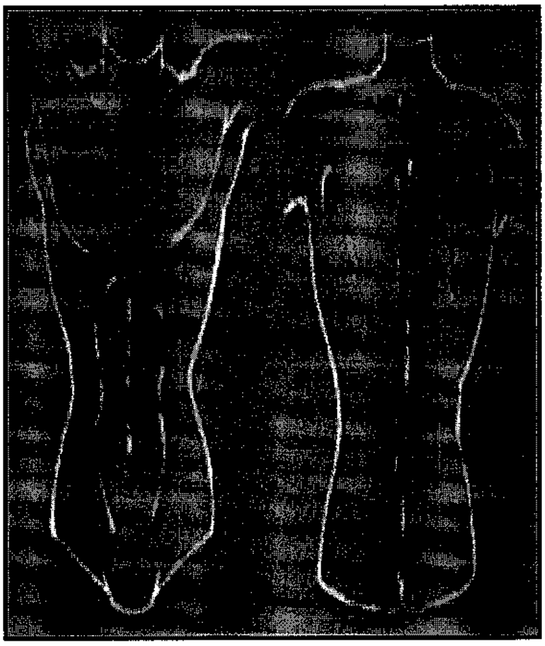


Figure 2.11: Trunk region

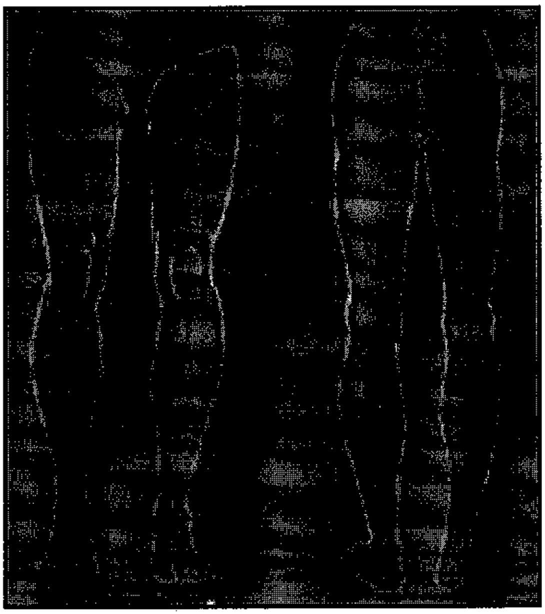


Figure 2.12: Legs region

2.2.1 Area

For each skin section, the amount of infected skin is measure as a percentage of the skin, it will score from 0 to 6 (refer to table 2.1 below). For example if the patient's head is 37% covered, the area score for his head-A_{head} would be 3.

Table 2.1: The area coverage score

Coverage	**Score**
0%	0
< 10%	1
10-29%	2
30-49%	3
50-69%	4
70-89%	5
90-100%	6

2.2.2 Severity

The severity is measured by three different parameters: Erythma (redness), Scaling and Thickness. The thickness of the skin can be measure since psoriatic skin is thicker than normal skin. Each of this parameter will be measured separately for each skin section. The score would be 0 to 4. Refer Table 2.2 below for severity score.

Table 2.2: Severity score

Severity	**Score**
None	0
Some	1
Moderate	2
Severe	3
Maximum	4

For example, if the patient's head thickness moderately, that would mean that T_{head} would be 2. If it's only some redness, his E_{head} score would be 1.The way to find others score are the same as above.

2.2.3 *Totaling Up the Index*

When all 16 of the above score are figured out, the PASI score for that patient is calculated. For each skin section, add up the three severity scores, multiply the total by the area score, and then multiply that result by the percentage of skin in that section, as follow :

Head: $(E_{head} + S_{head} + T_{head}) \times A_{head} \times 0.1 = Total_{head}$

Arms: $(E_{arms} + S_{arms} + T_{arms}) \times A_{arms} \times 0.2 = Total_{arms}$

Trunk: $(E_{trunk} + S_{trunk} + T_{trunk}) \times A_{body} \times 0.3 = Total_{body}$

Legs: $(E_{legs} + S_{legs} + T_{legs}) \times A_{legs} \times 0.4 = Total_{legs}$

PASI score $= Total_{head} + Total_{arms} + Total_{trunk} + Total_{legs}$

This PASI will range from 0 (no psoriasis) to 72 (covered head-to-toe, with complete redness, scaling, and thickness).

2.2.3.1 *Example of PASI calculation*

Table 2.3: Example of patient's data used for PASI calculation

		Data collection			
		Head	Arms	Trunk	Legs
1	**Redness**	3	1	3	1
2	**Thickness**	1	4	2	4
3	**Scaliness**	2	3	1	2
5	**Affected area (%)**	15	5	25	40
6	**Area score**	2	1	2	3
7	**Area weights**	0.1	0.2	0.3	0.4

Head: $(E_{head} + S_{head} + T_{head}) \times A_{head} \times 0.1 = Total_{head}$

Arms: $(E_{arms} + S_{arms} + T_{arms}) \times A_{arms} \times 0.2 = Total_{arms}$

Trunk: $(E_{trunk} + S_{trunk} + T_{trunk}) \times A_{body} \times 0.3 = Total_{body}$

Legs: $(E_{legs} + S_{legs} + T_{legs}) \times A_{legs} \times 0.4 = Total_{legs}$

PASI score = $Total_{head} + Total_{arms} + Total_{trunk} + Total_{legs}$

Based on the data from Table 2.3, the PASI score can be calculated as below:

Head: $(3 + 1 + 2) \times 2 \times 0.1 = 1.2$

Arms: $(1 + 4 + 3) \times 1 \times 0.2 = 1.6$

Trunk: $(3 + 2 + 1) \times 2 \times 0.3 = 3.6$

Legs: $(1 + 4 + 2) \times 3 \times 0.4 = 8.4$

PASI score = 14.8

The PASI total score is range from 0 to 72. If the PASI total score is greater than 12, it will consider as severe psoriasis. In clinical trials, the PASI total score generally fold between 0 to 24 [10].

2.3 MATLAB 7.0

In this project, the group will use MATLAB version 7.0 to process the images. The Image Processing Toolbox provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. The image can be enhance to improved intelligibility, extract features, analyze shapes and textures. Most toolbox functions are written in the open MATLAB 7.0 language. We can use the algorithms, modify the source code, and create our own custom functions to be use in the project.

2.3.1 Grayscale Intensity Image Conversion

An intensity image is data matrix, I whose values represent intensities within some range. MATLAB 7.0 stores and intensities image as a single matrix, with each element of the matrix corresponding to one image pixel. The matrix can be class double, uint8, or uint16.While intensity images are rarely saved with a colormap, MATLAB 7.0 uses a colormap to display them.

The elements in the intensity matrix represent various intensities, or gray levels, where the intensity 0 usually represents black and the intensity 1, 255, or 65535 usually represent full intensity, or white.

2.3.2 Graphical User Interface (GUI) of MATLAB 7.0

Graphical User Interface (GUI) is a program interface that takes advantages of the computer's graphics capabilities to make the program easier to use. Well designed graphical user interface can free the user from learning complex command languages. On the other hand, many users find that work more effectively with a command-driven interface, especially if they already know the command language [11].

Graphical user interfaces (GUI) have the following basic components:

2.3.2.1 Pointer

A symbol that appears on the display screen and use to select object and command. Usually, the pointer appears as a small angled arrow.

2.3.2.2 Pointing device

Pointing device is such a mouse or trackball which enable users to select the object on the display screen.

2.3.2.3 Icon

An icon is small pictures that represent command, files, or windows. By moving the pointer to the icon and pressing a mouse button, command can be execute or convert the icon into the window.

2.3.2.4 Desktop

The area on the display screen where icons are grouped.

2.3.2.5 Menus

Most graphical user interface can execute command by selecting a choice from a menu.

2.4 Theory

2.4.1 Image segmentation

Image segmentation is one of the most important steps leading to the analysis of processes image data. The segmentation produced a binary image used to check each individual pixel whether it belongs to an object of interest or not. A pixel has the value one if it is belongs to the object; otherwise it is zero. Segmentation is operated at the threshold between low-level image processing and the operations which analyze the shape of objects. In this project, several type of segmentation process has been used such as the thresholding technique and edge detection technique. All this technique plays the important role in completing this project, below are the brief description of the techniques used.

2.4.1.1 Thresholding

Thresholding is a method of image segmentation. Because of its intuitive properties and simplicity of implementation, image thresholding enjoys a central position in applications of image segmentation.

Thresholding in simple words means extracts objects from background. For an image (y, x) composed of light objects on a dark background, a threshold T can separates these two dominant modes. Then, any point (x, y) for which $f(x, y) > T$ is called an object point; otherwise the point is called a background point.

The most common form of image thresholding makes use of pixel grey level. Grey level thresholding applies to each pixel, the rules are,

$$g(x,y) = 0 \quad f(x,y) < T$$
$$g(x,y) = 1 \quad f(x,y) > T$$

where T is threshold value.

2.4.1.2 Edge based segmentation

An edge by definition is a boundary between two regions relatively uniform intensity, indicated by strong gradient or discontinuity in the intensity function. According to this definition, the intensity profile in a direction perpendicular to the edge will behave as a step function. In the direction along the edge, the intensity can vary smoothly. Edges that correspond to a strong second-order derivative, such as "peaks," will be detected as two edges according to the definition. Edge based segmentation represents a large group of methods based on information in the image. It is basically rely on edges found in an image by edge detecting operators such as Roberts operator, Sobel operator and Canny operator.

Roberts Edge Detector

Brief Description

The Roberts operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. It thus highlights regions of high spatial gradient which often correspond to edges. In its most common usage, the input to the operator is a grayscale image, as is the output. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point [12].

How It Works

In theory, the operator consists of a pair of 2×2 convolution masks as shown in Figure 2.6.One mask is simply the other rotated by 90°.

+1	0
0	-1

G_x

0	+1
-1	0

G_y

Figure 2.13: Roberts Cross convolution masks

These masks are designed to respond maximally to edges running at 45° to the pixel grid, one mask for each of the two perpendicular orientations. The masks can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G| = \sqrt{(G_x^2 + G_y^2)}$$

Although typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y|$$

which is much faster to compute.

The angle of orientation of the edge giving rise to the spatial gradient (relative to the pixel grid orientation) is given by:

$$\theta = \arctan(G_y / G_x) - 3\Pi/4$$

In this case, orientation 0 is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image, and other angles are measured anticlockwise from this. Often, the absolute magnitude is the only output the user sees, the two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator shown in Figure 2.7.

P_1	**P_2**
P_3	**P_4**

Figure 2.14: Pseudo-Convolution masks used to quickly compute approximate gradient Magnitude

Using this mask the approximate magnitude is given by:

$$|G| = |P_1 - P_4| + |P_2 - P_3|$$

Sobel Edge Detector

Brief Description

The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial gradient that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image [12].

How It Works

In theory at least, the operator consists of a pair of 3×3 convolution masks as shown in Figure 2.8. One mask is simply the other rotated by 90°. Cross operator;

-1	**0**	**+1**
-2	**0**	**+2**
-1	**0**	**+1**

G_x

+1	**+2**	**+1**
0	**0**	**0**
-1	**-2**	**-1**

G_y

Figure 2.15: Sobel convolution masks

These masks are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one mask for each of the two perpendicular orientations. The masks can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G| = \sqrt{(G_x^2 + G_y^2)}$$

Although typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y|$$

which is much faster to compute.

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \arctan(G_y / G_x) - 3\Pi/4$$

In this case, orientation 0 is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image, and other angles are measured anticlockwise from this. Often, this absolute magnitude is the only output the user sees, the two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator shown in Figure 2.9.

P1	**P2**	**P3**
P4	**P5**	**P6**
P7	**P8**	**P9**

Figure 2.16: Pseudo-convolution masks used to quickly compute approximate gradient Magnitude

Using this mask the approximate magnitude is given by:

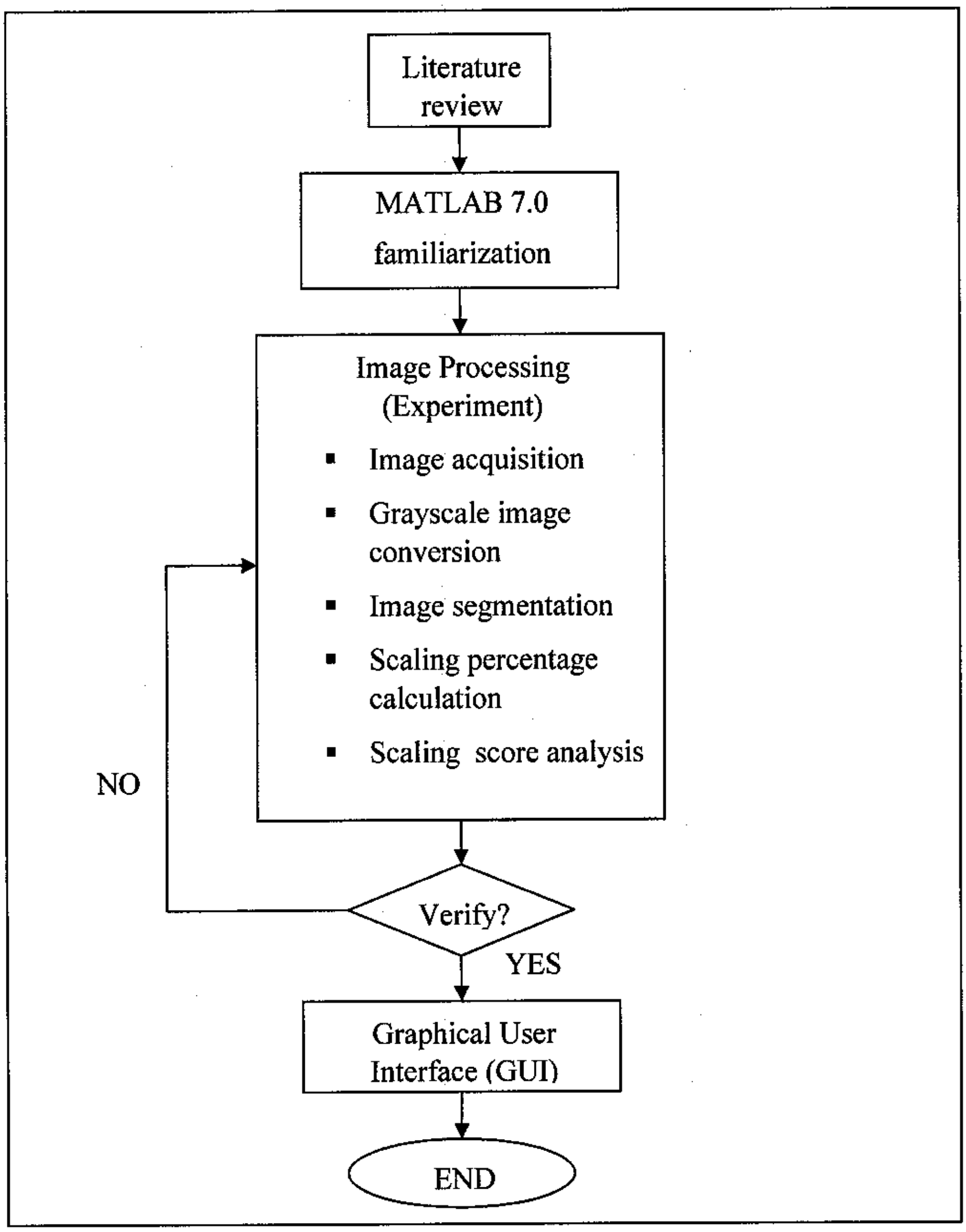
$$|G| = \left|\left(P_1 + 2\times P_2 + P_3\right) - \left(P_7 + 2\times P_8 + P_9\right)\right| +$$

$$\left|\left(P_3 + 2\times P_6 + P_9\right) - \left(P_1 + 2\times P_4 + P_7\right)\right|$$

CHAPTER 3
METHODOLOGY

3.1 Procedure Identification

Basically, there are a few steps of systematic approach for this project which is represented by block diagram below:



3.1.1 Literature review

Research is very useful especially in assisting the successfully of the project. The researches have been done during the early stage of the project. Literature from the internet is the main source in getting all the latest technology of information related to the project instead of getting resource from the books at the library.

3.1.2 MATLAB familiarization

In basic design, MATLAB 7.0 code was used intensively for creating several commands necessary for this project. The main area the group must familiarized is the digital image processing. The area covered including how to load an image, grayscale converting, image segmentation which is thresholding and edge detection techniques, area of object calculation and also Graphical User Interface (GUI) implementation. After we become familiar with the MATLAB 7.0 software, the next stage is constructing the MATLAB 7.0 algorithm .This is the most crucial part since it is a complex procedure.

3.1.3 Image Processing Toolbox

This stage is parted into five sections:-

i. Image acquisition
ii. Grayscale image conversion
iii. Image segmentation
iv. The area of object calculation
v. Scaling analysis

3.1.3.1 Image acquisition

At the beginning of the project, the image data is downloaded from the internet resource. Since the actual image of psoriasis infected skin is a confidential and scarce resource, we run the experiment using this downloaded image.

3.1.3.2 Grayscale image conversion

In converting RGB or colormap image to grayscale, function used is RGB2GRAY.It convert RGB images to grayscale by eliminating the hue and saturation information while retaining hue luminance. If the input is an RGB image, it can be class uint8, uint16 or double; the output image I is the same class as the input image. If the input is a colormap, the input and output colormaps are both of class double. The general syntax used is

Image2 = RGB2GRAY (Image);

Original input image is in RGB or colormap format, to analyze this image it must be converting to grayscale image using MATLAB. Figure 3.1 below show the result of grayscale conversion:

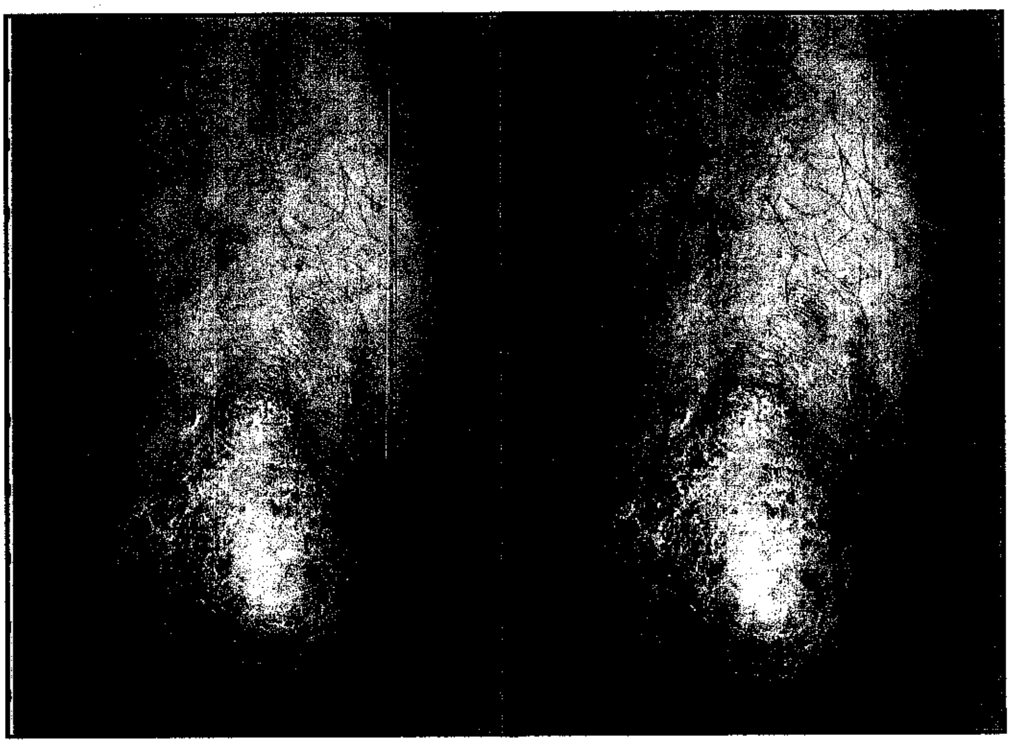


Figure 3.1: Image being converts from RGB to grayscale image

3.1.3.3 Image segmentation

Two type of image segmentation is used in these projects which are edge detection method and thresholding method. Both methods are combined together in a function called Sobel edge detector, Roberts edge detector or Canny edge detector.

Sobel Method

BW = edge (I,'sobel') specifies the Sobel method.

BW = edge (I,'sobel', thresh) specifies the sensitivity threshold for the Sobel method. 'edge' ignores all edges that are not stronger than thresh. If the 'thresh 'is not specify, or if 'thresh' is empty ([]), edge chooses the value automatically.

BW = edge (I,'sobel', thresh, direction) specifies the direction of detection for the Sobel method. 'direction' is a string specifying whether to look for 'horizontal' or 'vertical' edges or 'both' (the default).

[BW, thresh] = edge (I,'sobel'...) returns the threshold value edge detector.

Roberts Method

BW = edge (I,'roberts') specifies the Roberts method.

BW = edge (I,'roberts',thresh) specifies the sensitivity threshold for the Roberts method. 'edge' ignores all edges that are not stronger than thresh. If the 'thresh' is not specify, or if 'thresh' is empty ([]), 'edge' chooses the value automatically.

[BW, thresh] = edge (I,'roberts'...) returns the threshold value.

Canny Method

BW = edge (I,'canny') specifies the Canny method.

BW = edge (I,'canny', thresh) specifies sensitivity thresholds for the Canny method. 'thresh' is a two-element vector in which the first element is the low threshold, and the second element is the high threshold. If a scalar is used for thresh, this value is used for the high threshold and 0.4*thresh is used for the low threshold. If the thresh is not specify, or if thresh is empty ([]), edge chooses low and high values automatically.

BW = edge (I,'canny', thresh, sigma) specifies the Canny method, using 'sigma' as the standard deviation of the Gaussian filter. The default sigma is 1; the size of the filter is chosen automatically, based on 'sigma'.

[BW, thresh] = edge (I,'canny',...) returns the threshold values as a two-element vector.

Image segmentation implementation

The edge detection and the thresholding technique will be applied to the grayscale image. This edge detector will performs a simple, quick to compute, 2-D spatial gradient measurement on an image. It therefore highlights regions of high spatial gradient which correspond to edges. The input to the operator is a grayscale image, as is the output. Pixel values at each point in the output will represent the estimated absolute magnitude of the spatial gradient of the input image at that point. For this project, the edges will correspond to the boundaries and membrane of the scaling texture with respect to the psoriasis diseases. The thresholding will do automatically right after the edge detection process and the result is as Figure 3.2:

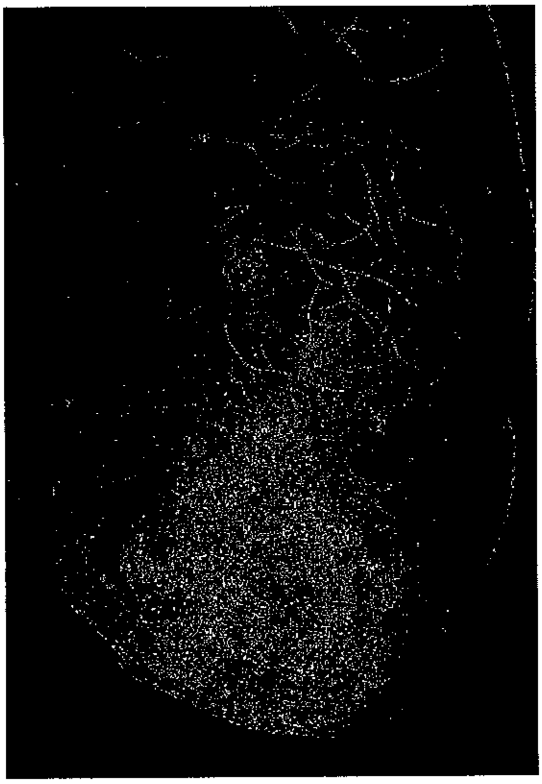


Figure 3.2: Result of image after the edge detection and thresholding technique is applied

3.1.3.4 The area of object calculation

This project get the scaling value of the infected skin by calculate its pixel percentage. The pixel percentage is get by compare the pixel of scale produce after applying the edge detection process with the total pixel of that image.

Total = bwarea (BW) estimates the area of the objects in binary image BW.

'Total' is a scalar whose value corresponds roughly to the total number of on pixels in the image, but might not be exactly the same because different patterns of pixels are weighted differently. In this project, the total number of pixels calculated after the edge detection process is corresponds to the total scaling score. Hence, to get the scaling percentage, it can be computed using equation below:

$$Scalingpercentage = \frac{Totalscalingscore(Pixels)}{Totalsizeofimage(Pixels)} \times 100\%$$

3.1.3.5 *Scaling analysis*

From the result of scaling percentage of infected skin, the scaling score for PASI can be obtained. The scaling score is ranged from 0 to 4 depend on how severed the diseases is. Scaling percentage of infected skin (x %) is obtained by dividing the total pixels of scaling calculated over the total size of pixels of that image. It is mean, if the scaling of the infected skin is bad, the scaling percentage will score high resulting to the higher PASI scaling score. See Table 3.1 below for the details of the scaling percentage and its corresponding scaling score.

Table 3.1: Scaling percentage and its corresponding scaling score

Scaling percentage of infected skin x %	**Severity**	**PASI Scaling score**
x <= 0.555	None (Normal Skin)	0
0.555 < x <= 2	Some	1
2 < x <= 3	Moderate	2
3 < x <= 4	Severe	3
x > 4	Maximum	4

3.1.4 Creating a Graphical User Interface (GUI)

Several steps have been followed to create the desired Graphical User Interface (GUI).This GUI will help the user in accessing the built program. Below are the procedure has been used to come out with the desired result:

3.1.4.1 Set up the figure of GUI layout editor and its function

The Figure 3.3 below shows the set up of the components use in the program. The components are selected from the components palette located at left side of the layout. The property inspector for each component is used to set the required property for each component. Then the program is inserted into each callback functions for the respective component. The program has been built in the early process.

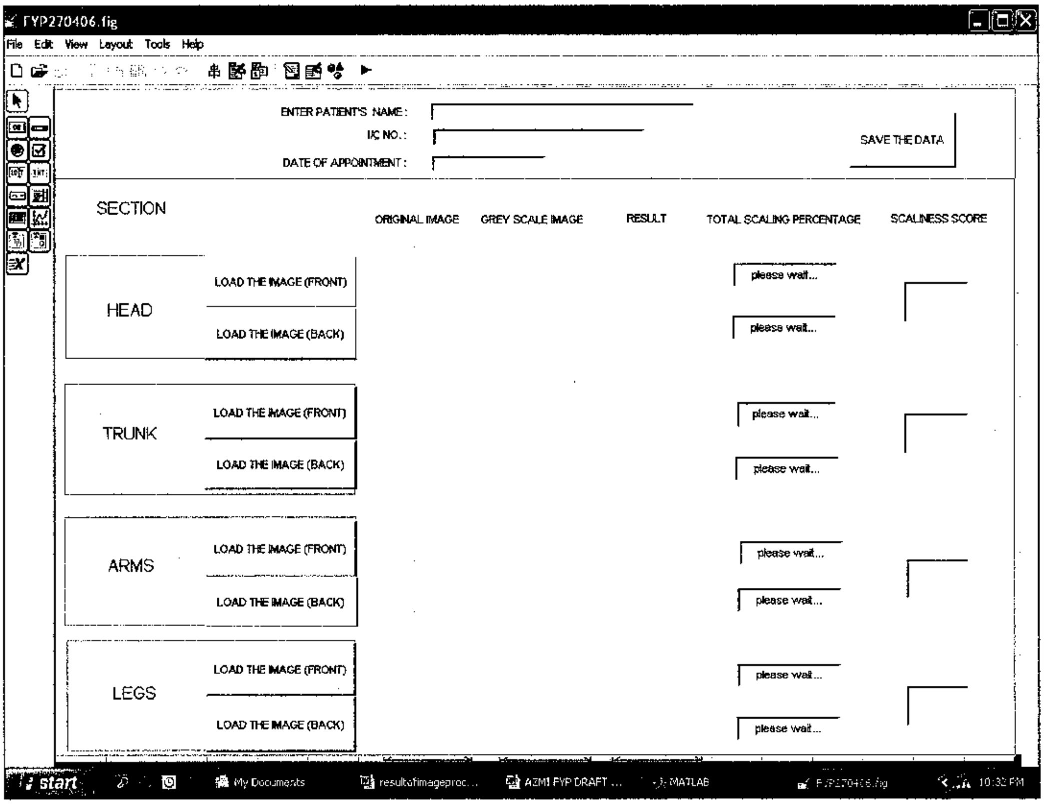


Figure 3.3: The GUI layout editor

3.1.4.2 *Run the GUI*

To run a GUI, Run is select from the Tools menu, or the run button is click on the toolbar. The following Figure 3.4 shows the functioning GUI outside the Layout Editor.

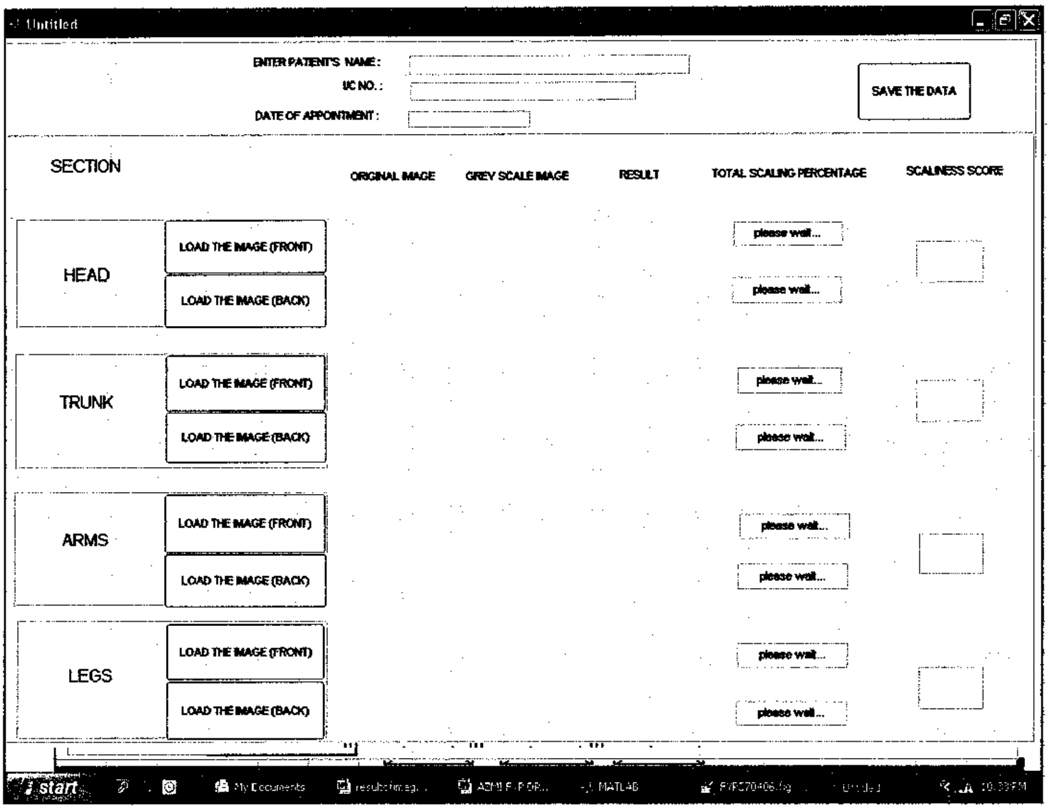


Figure 3.4: The functioning GUI outside the Layout Editor

3.1.4.3 *Load the image of Psoriasis plague*

Figure 3.5 below show the image is being loaded into the GUI. It is done by clicking the 'Load the image' icon in that GUI. The loading processes are divided into four sections which are the head, trunk (body), arms and legs section. Each section will process two images .For examples in head section; it will process the front side and the back side of head image. This program will load, process and analyze each images .Lastly it will display the result from that image in term of total scaling percentage and also the scaliness score. Figure 3.6 shows the completed process together with their result.

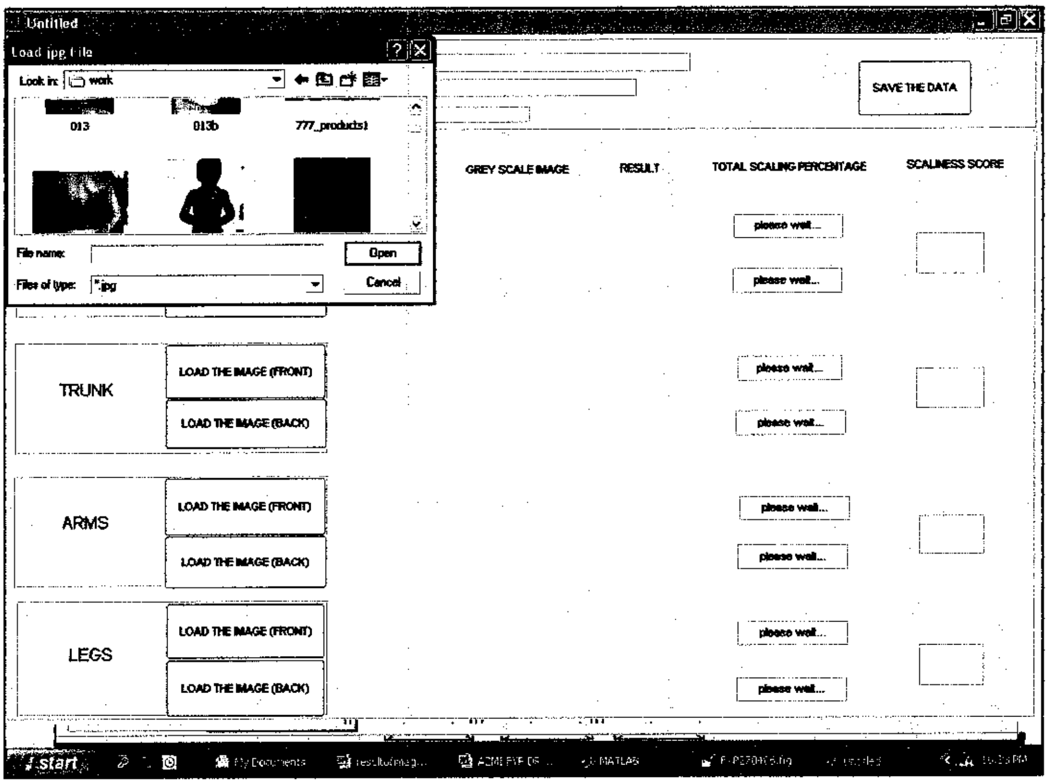


Figure 3.5: The image being loaded into the GUI

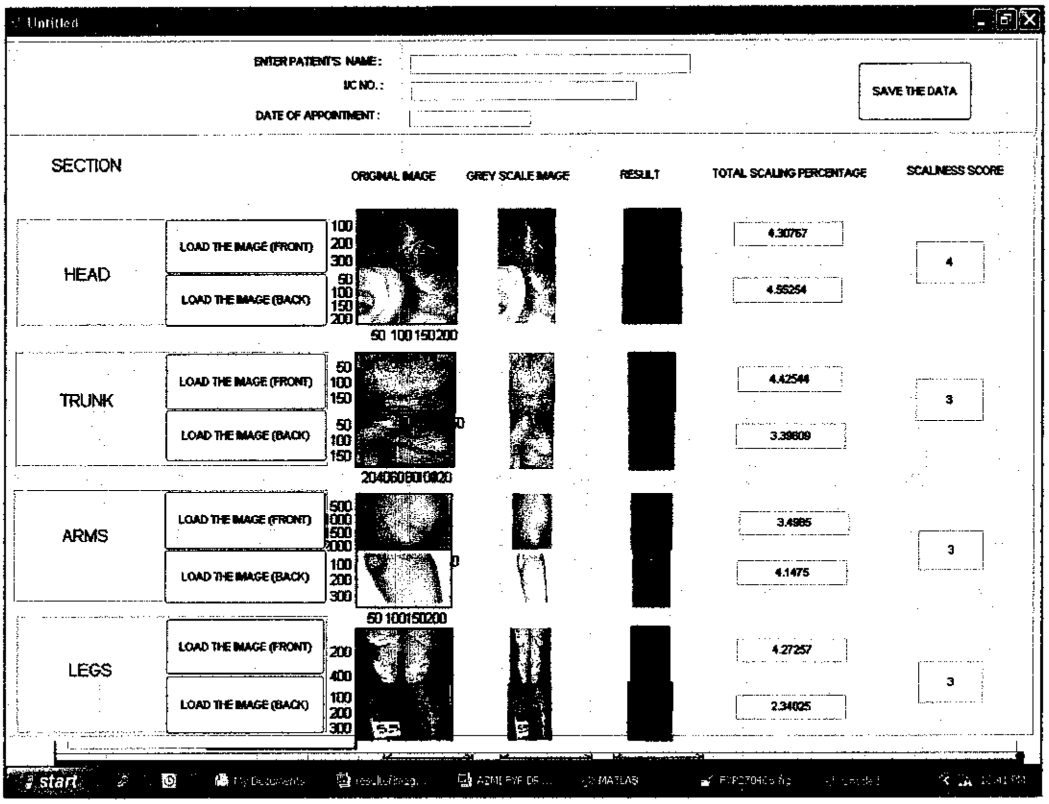


Figure 3.6: The complete image loading together with their result

3.1.4.4 Saving the data

Lastly, the information of that patient is inserted and the data will be saved .This data will be used to calculate the total PASI score for that patient. This result will help the dermatologist to evaluate the severity of that psoriasis patient and also will help them by giving an appropriate treatment and medication. The saving process is shown as Figure 3.7 below:

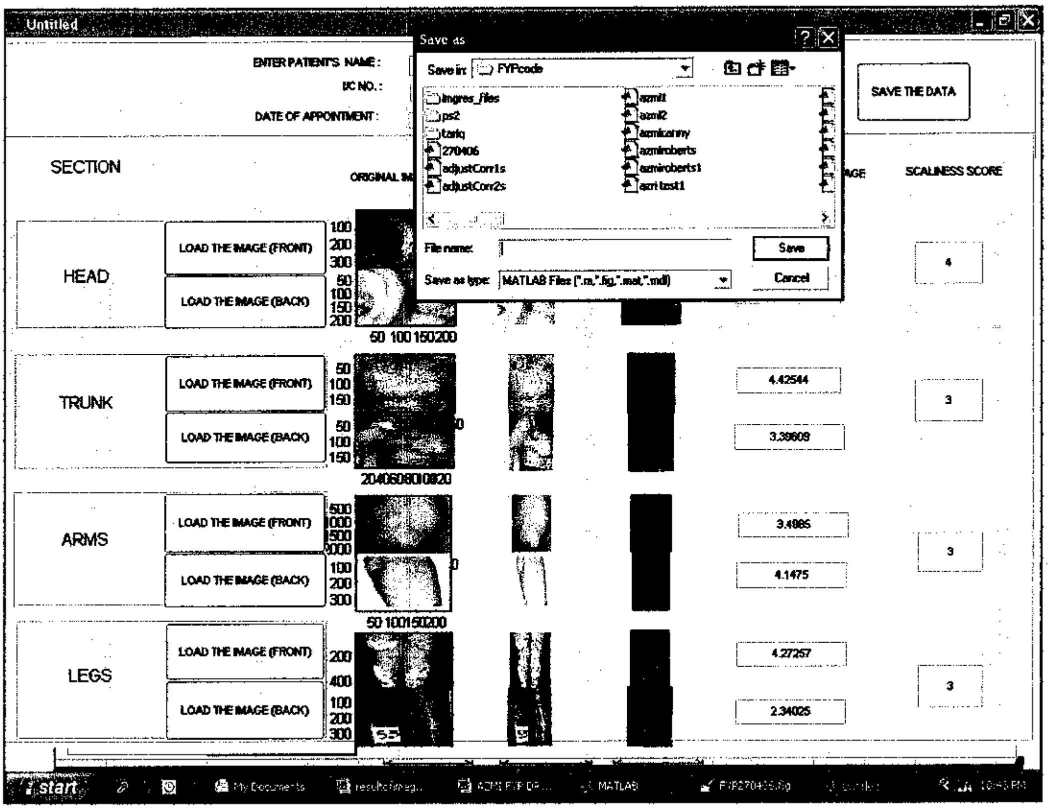


Figure 3.7: The result being saved

3.2 Tools

Tools used for this project are:-

i. MATLAB 7.0 software
ii. Adobe Photoshop software
iii. Psoriasis Plague images
iv. Digital camera
v. Workstation
vi. Personal Computer (PC)

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Results

4.1.1 Results output

Below are the results of the processed image obtained from the output of the program. Each dimension and size of images is different because these images are taken from the internet resource.

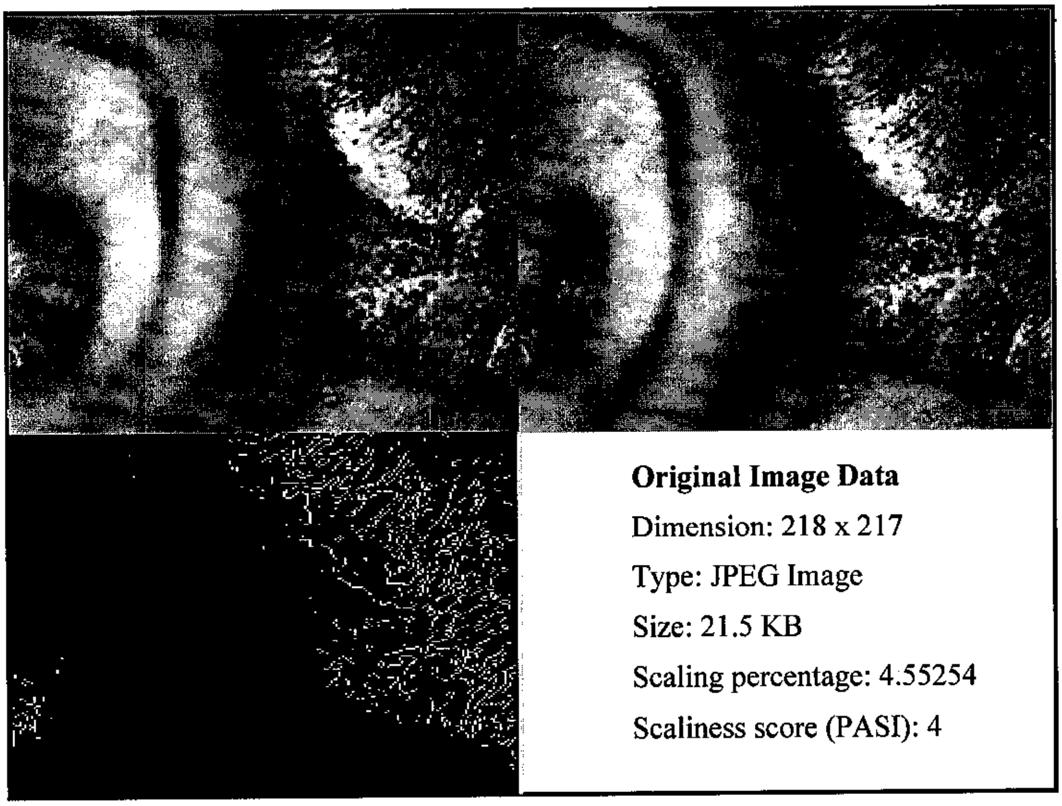


Figure 4.1: Result for head section

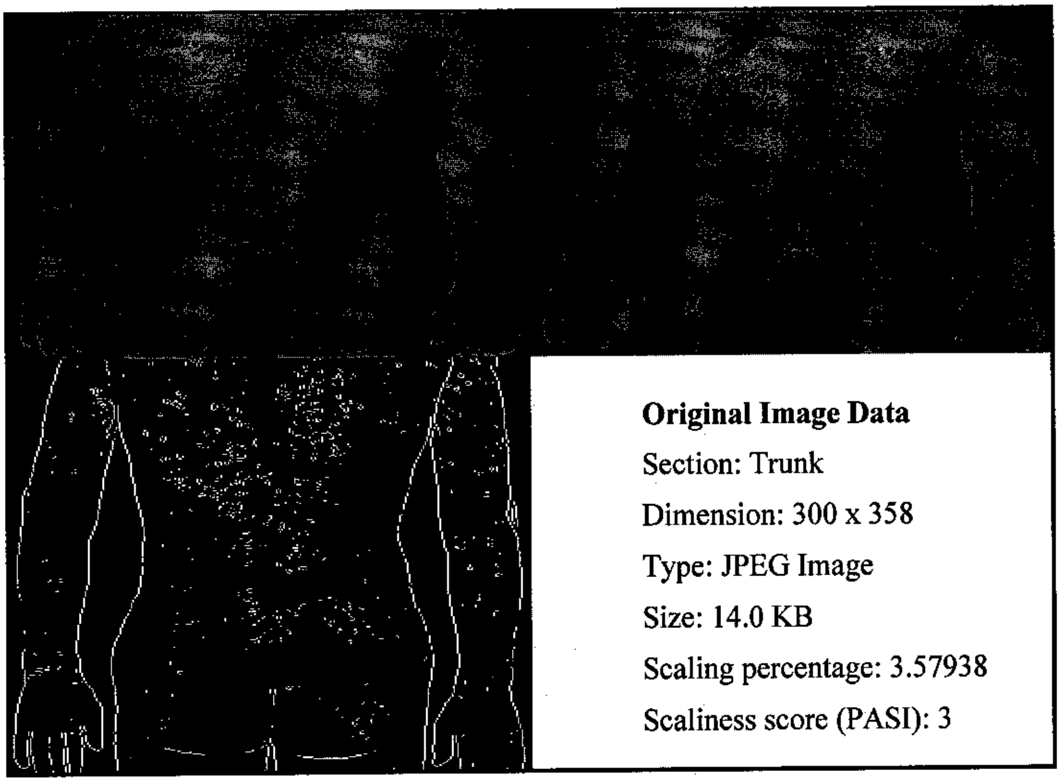


Figure 4.2: Result from trunk section

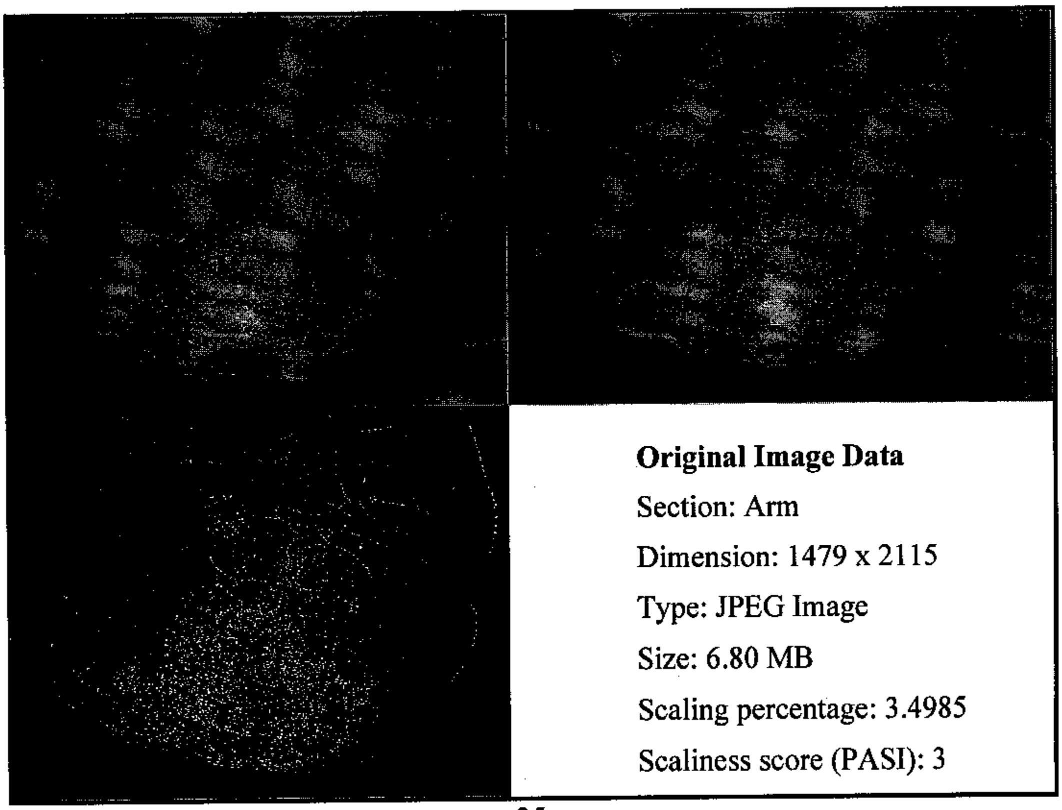


Figure 4.3: Result from arm section

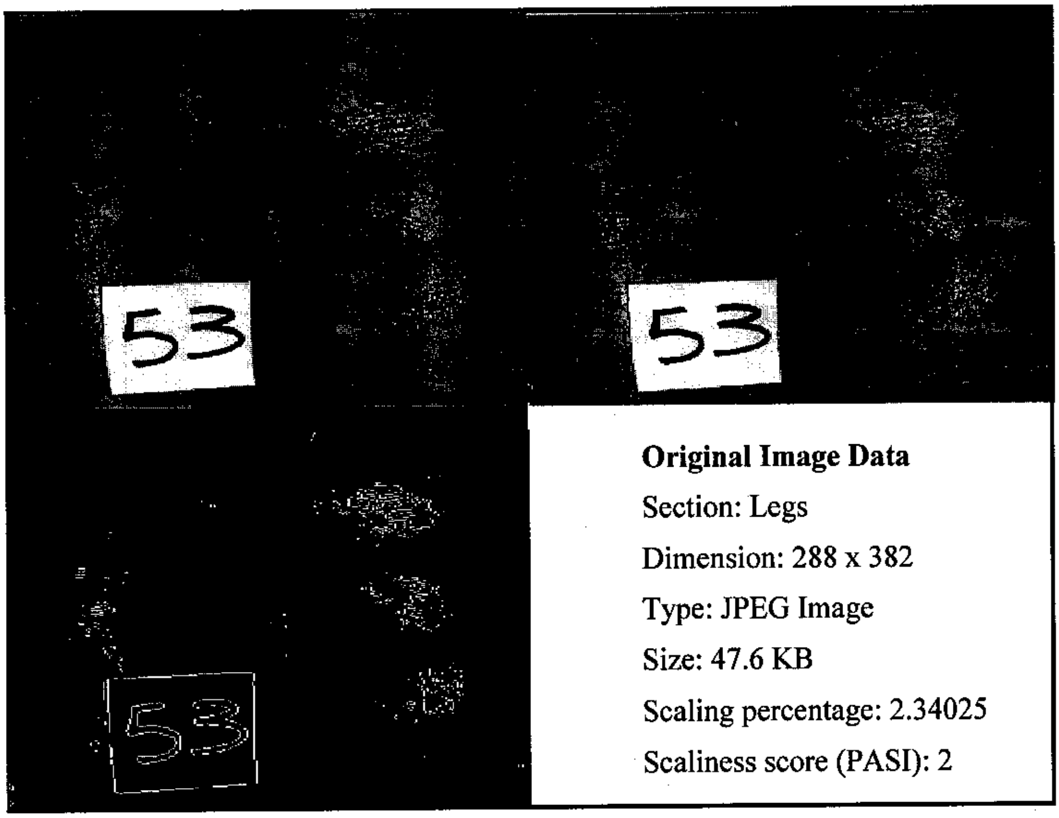


Figure 4.4: Result from legs section

4.2 Discussion

From the output results of the project, it is proved that an image based system to objectively score the degree of scaling in psoriasis lesions is successfully done. The results effectively showed the individual scaling score for Psoriasis Area and Severity Index (PASI) calculation. The score is range from 0 to 4 depending on how severe the psoriasis is. In completing the project, there are some limitation regarding to the input data because it is not like what the input of the program suppose to be, hence an errors occur. This error will contribute to the uncertainties results compute by the program. So a standard procedure must be followed to encounter such error and to ensure an accurate result is obtained. In this section, the problem encounter during the project is discussed and the suggested improvement is proposed here:

4.2.1 Capturing the image

The image input of this program is different in term of quality, dimension and size, hence the output also differ compare to each other. The reason why this image has been taken for implementation is because there are no images exactly like what we want that available at our area also in any website in the internet. Supposedly, the image inputs the program should be are as Figure 4.5 below:

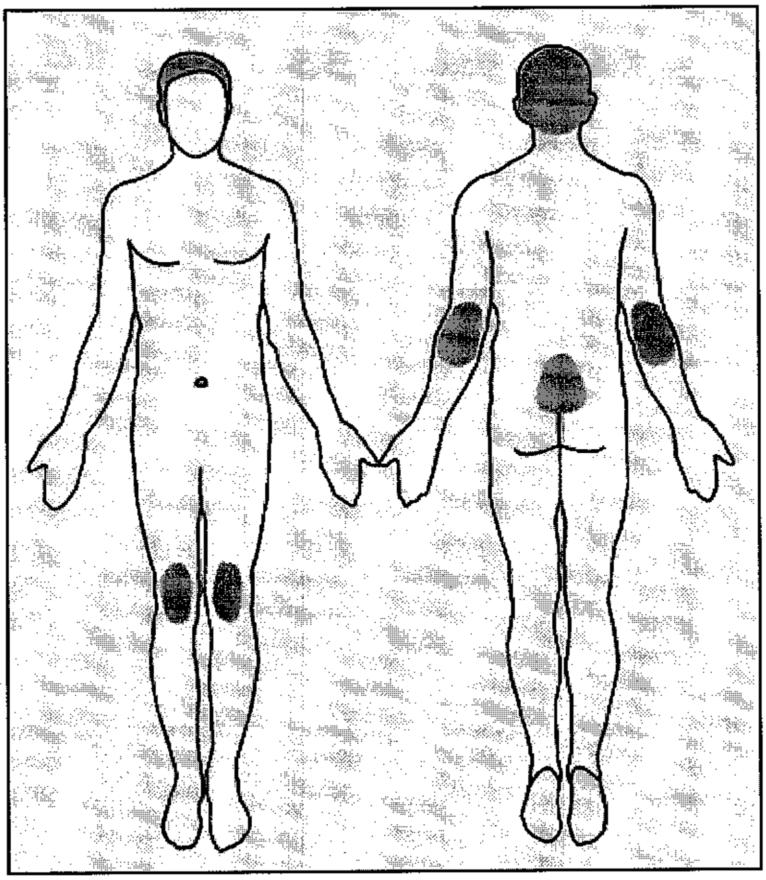
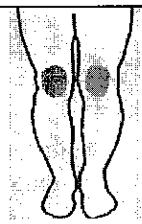


Figure 4.5: The patient orientation while capturing the image for use as input

Two images will be captured from a psoriasis patient .One from the front side of the patient and the other one is from the rear side of the patient (Figure 4.5). The editing process will be prepared using Adobe Photoshop 7.0 software. This software is used to separate that image into four sections which are head, trunk, arms and legs section. The result after the editing process should be as Table 4.1:

Table 4.1: The required input image for the program

SECTION	IMAGE SIDE	IMAGE POSITION
HEAD	FRONT	
	BACK	
TRUNK	FRONT	
	BACK	
ARMS	FRONT	
	BACK	
LEGS	FRONT	
	BACK	

4.2.2 Image distance

To provide a standardized result, the distance between the patient and the camera should be fixed. This arrangement is important to ensure the image ratio between the patient body and the background of image is uniform and not varies compare to the other patient. This procedure also will ensure the result among different patient is standardized. The set up of camera and patient orientation is shown in Figure 4.6 below.

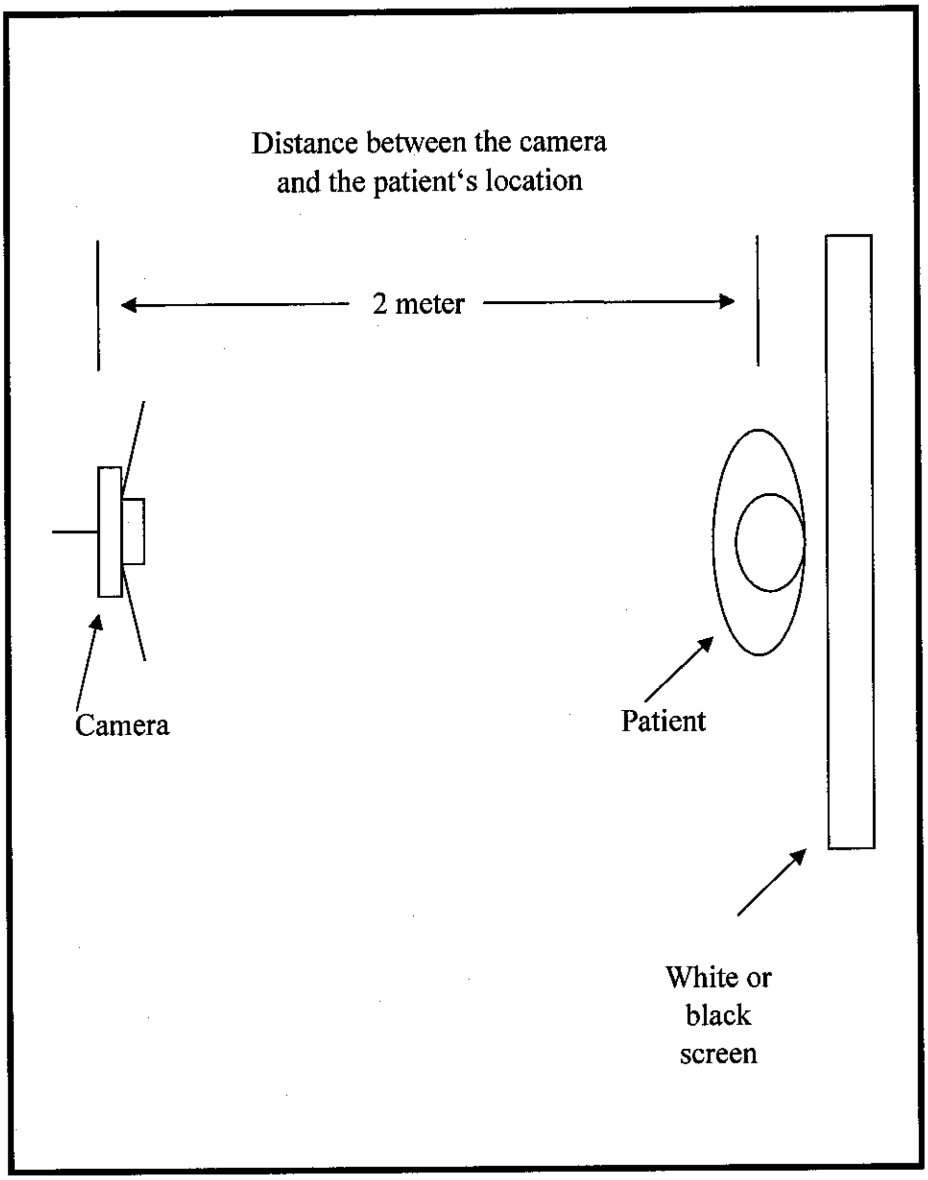


Figure 4.6: Set up of camera and the patient's orientation

4.2.3 Background of image captured

To ensure the result of the program is more accurate, the background of image captured should be uniform color (white or black color is recommended) and no scaly surface appear on it. This is done to avoid any uncertainties occur while calculating the scaling percentage. The following figure shows the example of uncertainty that occur due to appearance of the grain surface on the background.

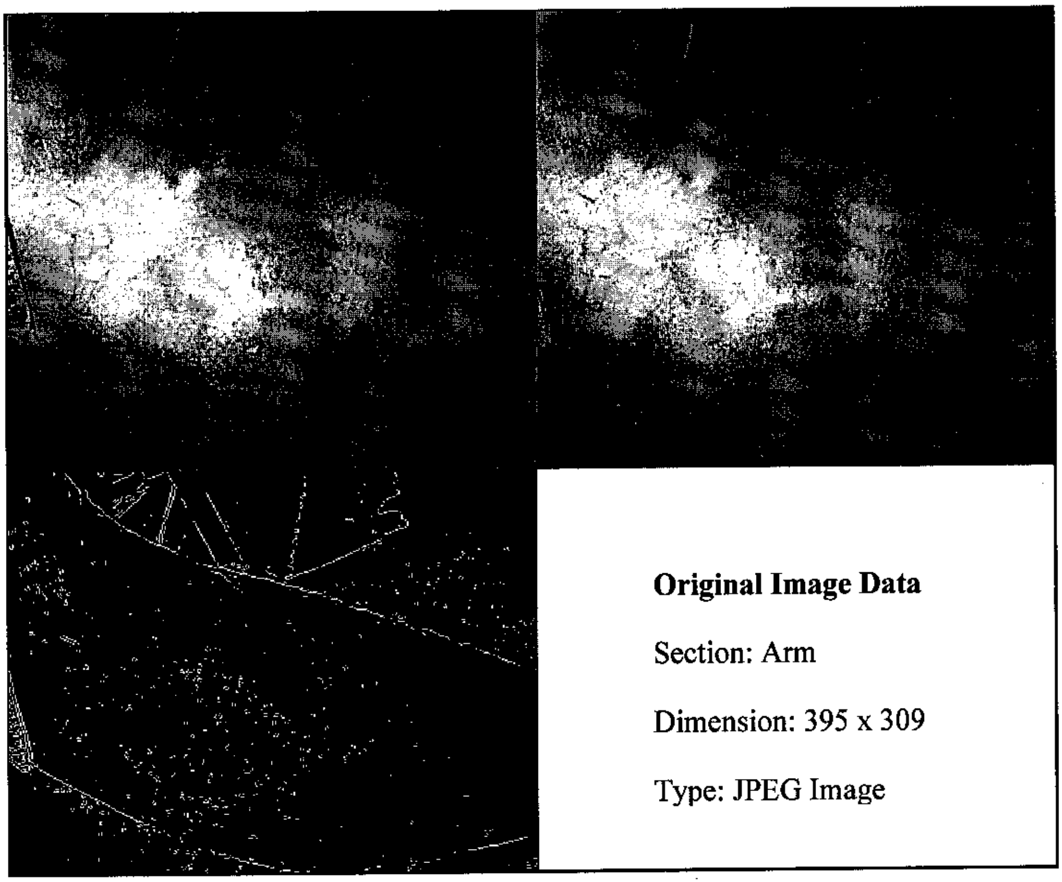


Figure 4.7: The result of process image with grain surface background

4.2.4 Image dimensions

From the results output, the author can conclude that the best result can be obtained from the image with the high dimension or high value of pixels. The dimension setting can be done at the digital camera .There are many types of image dimension available now depending on the camera used. This project implementation is based on a Nikon digital camera (CoolpixL2 model), there are several dimensions provide in this camera and the propose dimension is 2816×2112. This dimension is chosen because it showed the best results compare to the others. Moreover, it is the highest dimension provide in this camera. The following figures show the image result from the CoolpixL2 model camera with dimensions of 2816×2112.

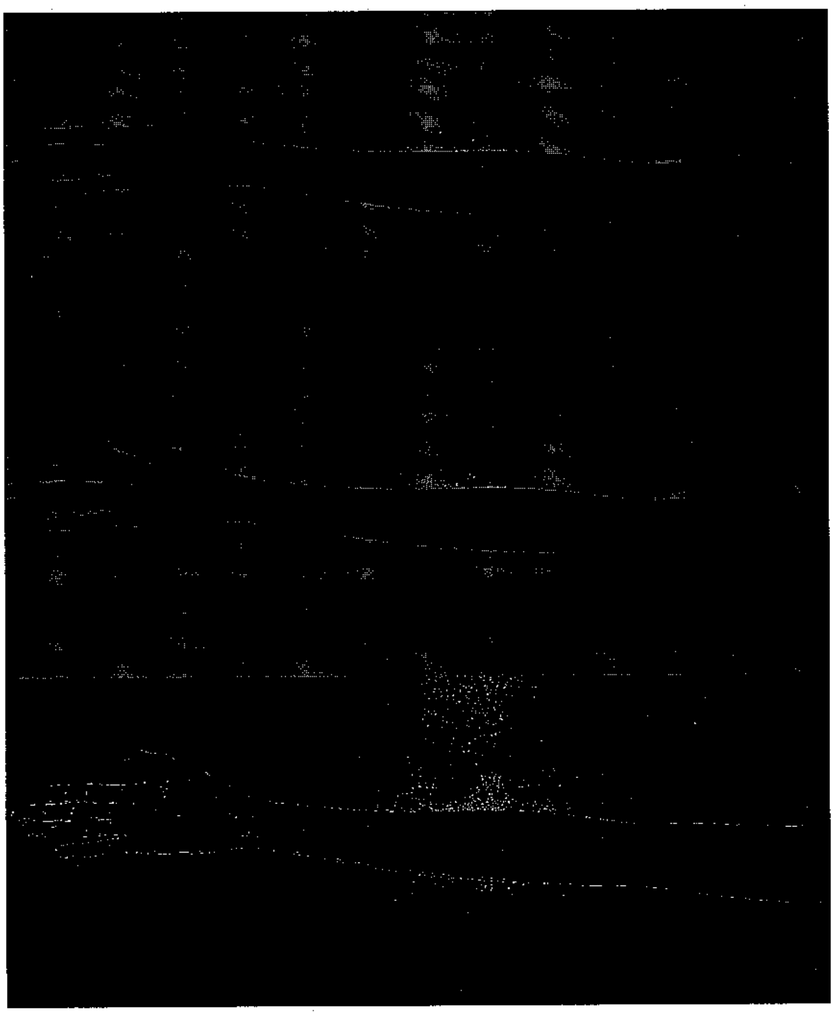


Figure 4.8: The resulting output when the image (normal skin) with dimensions of 2816×2112 is used as input

4.2.5 Scaliness score

Since the required image as the project needed is not available, there are some errors occur in the current program that we must take into consideration. This errors are made from the low quality of input images; the image is compressed from its original; the difference size among the input images and inconsistent position of image capturing. To improve the result's accuracy, the corresponding scaling score with respect to the scaling pixel calculated need to be defined again. It must be done right after the desired input image is obtained. Besides, the project implementation should be complete under supervision and guidance of the dermatologist. In this way, the resulted outputs are believed to be more accurate and similar to the doctor's result.

CHAPTER 5
RECOMMENDATION

There are some parts that can be improved to produce more accurate result. Further analysis can be done if the project is extended and the project scope is broadened.

5.1 Input data

To get more accurate result, the real image of psoriasis skin diseases is needed. The real image is different compare to the one from internet resource because it is not alter yet from its original. This image can be get from the clinicians. The image should be capture from the high quality of digital camera; this is done to ensure the quality of image is good and free from any noise, blurry or shadowy.

5.2 Image capturing technique

The position of patient while capturing their image must be standardized .The distance of camera and the patient should be fixed and the lighting must be in the right position. The background of the image also should be one color only, for example the background color is black.

5.3 Project improvement

This project is applied by using the MATLAB 7.0 software only, hopefully in the future it will be applied to the other software, for example C++ programming. So that, the result can be compare and the system improvement can be done.

CHAPTER 6
CONCLUSION

For the conclusion, the objective of this Final Year Project is reached. The objectives are to build an image based system to objectively score the degree of scaling in psoriasis lesions disease and calculate the Psoriasis Area and Severity Index (PASI) on patient; to develop a cheap and practical technique of processing, comparing and analyzing skin disease to aid the clinical diagnosis; and to present an efficient method for the dermatological images characterization using MATLAB 7.0. The program can be further modified to ensure its smooth running in the future.

BIBLIOGRAPHY

Ganster H, Pinz P, Rohrer R, Wildling E, Binder M, Kittler H. Automated melanoma recognition. *IEEE Transactions on Medical Imaging,* 20:233-239. doi: 10.1109/42.918473, 2001 .

Kjoelen M, Thompson M, Umbaugh S, Moss R, Stoecker W. Performance of Artificial Intelligence Methods in Automated Detection of Melanoma. *IEEE Engineering Medicine and Biology*, 14:411, 416. doi: 10.1109/51.395323,1995.

Maglogiannis. I. Automated Segmentation and Registration of Dermatological Images. *Proceedings of the 2002 International Conference on Parallel and Distributed Processing Techniques and Applications,* USA, pp. 121-126, 2002.

Umbaugh SE, Wei Y, Zuke M. Feature Extraction in Image Analysis. IEEE *Engineering in Medicine and Biology*, 16:62-73. doi: 10.1109/51.603650, 1997.

Hall PN, Claridge E, Smith M. Computer screening for early detection of melanoma , is there a future?. *Br J Dermatol*; 132:325-338, 1995.

Hansen G, Sparrow E, Kokate JY, Leland KJ, Iaizzo PA. Wound status evaluation using color image processing. *IEEE Transactions on Medical Imaging*; 16:78-86. doi: 10.1109/42.552057,1997.

Moncrieff M,Cotton S,Claridge E,Hall P. Spectrophotometric Intrcutaneous Analysis-A New Technique For Imaging Pigmented Skin Lesions. *British Journal of Dermatology,* 146(3), Pages: 448-457, 2002.

Harald Ganster, Axel Pinz, Reinhard R\hrer, Ernst Wildling,Michael Binder, and Harald Kittler. Automated Melanoma Recognition. *IEEE Transactions On Medical Imaging,* Vol. 20, No. 3, March 2001.

Brett D. Thackray, Alan C. Nelson Semi-Automatic Segmentation of Vascular Network Images Using a Rotating Structuring Element (ROSE) with Mathematical Morphology and Dual Feature Thresholding. *IEEE Transactions on Medical Imaging,* Vol. 12, No. 3, September 1993.

Jong Kook Kim, Hyun Wook Park. Statistical Textural Features for Detection of Microcalcifications in Digitized Mammograms. *IEEE Transactions on Medical Imaging,* Vol. 18, No. 3, March 1999.

Qiang Ji, John Engel, Eric Craine. Texture Analysis for Classification of Cervix Lesions. *IEEE Transactions on Medical Imaging,* Vol. 19, No. 11, November 2000.

Guoliang Fan,Xiang-Gen Xia,. Wavelet-Based Texture Analysis and Synthesis Using Hidden Markov Models. IEEE *Transactions on Circuits and Systems-I: Fundamental Theory and Applications,* Vol. 50, No. 1, January 2003.

Jitendra Malik, Serge Belongie, Thomas Leung, Jianbo Shi. Contour and Texture Analysis for Image Segmentation. *International Journal of Computer Vision* 43(1), 7–27 Computer Science Division, University of California at Berkeley, Berkeley, 2001.

R. Rajesh, M.R. Kaimal , K. Srinivasan. A Note on Medical Image Analysis and Visualization using Matlab . *GVIP 05 Conference*, Egypt ,19-21 December 2005.

Autio, J., Luukkanen, S., Rantanen, L., and Visa, A. The Classification and Characterisation of Rock Using Texture Analysis by Co-occurrence Matrices and the Hough Transform. *International Symposium on Imaging Applications in Geology*, pp. 5-8, Belgium, May 6-7 1999.

Eva M. van Rikxoort. Texture analysis: *Graduate research proposal in AI,* April 15, 2004.

Image of Psoriasis plague available at:
http://images.google.com.my/

Article on, "An Introduction to Psoriasis Treatment "available at:
http://www.mydna.com/health/skin/psoriasis/overview/treatintro.html

REFERENCES

[1] This article can be accessed at:

http://www.manbir-online.com/psoriasis.htm

[2] PASI Calculator software, *PASI Educational Program, Psoriasis and Severity Index Toolkit*, Biogen Idec, Australia.

[3] This article can be accessed at:

http://www.wrongdiagnosis.com/p/psoriasis/subtypes.html

[4] S R Feldman, G G Krueger Ann Rheum," Psoriasis assessment tools in clinical trials," 64(Suppl II):ii65–ii68. doi: 10.1136/ard.2004.031237 ,2005.

[5] Prof Ir Dr Ahmad Fadzil M H & Farah Aini Nordin,Telehealthcare-Monitoring of Skin Diseases.

[6] Prof Dr Ahmad Fadzil M H & Thai Vinh Tuan,Content-Based Skin Image Retrieval.

[7] This article can be accessed at:

http://www.wrongdiagnosis.com/p/psoriasis/subtypes.html

[8] This article can be accessed at:

http://www.astronclinica.com/index.htm

[9] This article can be accessed at:

http://www.psorsite.com/orig_index.html

[10] SR Feldman. A quantitative definition of severe psoriasis for use in clinical trials. *Journal of Dermatological Treatment*; 15:27-29, 2004.

[11] Mastering MATLAB 6 "A comprehensive tutorial and reference" , *Duane Hanselman and Bruce Littlefield*, Prentice Hall, 2001.

[12] R. Fisher, S. Perkins, A. Walker and E. Wolfart. HIPR Hypermedia *Image Processing Reference*, Published by J. Wiley & Sons Ltd, 1996.

APPENDICES

APPENDIX A
PROJECT GANT CHART

First Semester of 2 Semester Final Year Project

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	**Selection of Project Topic**	■															
	Propose Topic																
	Topic assigned to students																
2	**Preliminary Research Work**		■	■													
	Introduction																
	Objective																
	List of references/literature																
	Project planning																
3	**Submission of Preliminary Report**			●													
4	**Project Work**				■	■	■	■									
	Reference/Literature																
	Matlab implementation																
5	**Submission of Progress Report**							●									
6	**Project work continue**								■	■	■	■					
	Matlab implementation																
7	**Submission of Interim Report Final Draft**												●				

8	**Submission of Interim Report**													●			
9	**Oral Presentation**														●		

Process ■

Suggested milestone ●

Second Semester of 2 Semester Final Year Project

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	**Project Work Continue**		■	■																		
	Matlab implementation																					
2	**Submission of Progress Report 1**				●																	
3	**Project Work Continue**				■	■	■	■														
	Matlab implementation																					
4	**Submission of Progress Report 2**								●													
5	**Project work continue**								■	■	■	■										
	Result analysis																					
6	**Submission of Dissertation Final Draft**																●					
7	**Oral Presentation**																					●
8	**Submission of Project Dissertation**																					●

Process ■

Suggested milestone ●

APPENDIX B
GUI PROGRAM

===

```
function varargout = FYP270406(varargin)


gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @FYP270406_OpeningFcn, ...
                   'gui_OutputFcn',  @FYP270406_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

function FYP270406_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = FYP270406_OutputFcn(hObject, eventdata, handles)
```

```
varargout{1} = handles.output;
```

```
% --- Executes on button press in pushbutton1.

 function pushbutton1_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile('*.jpg', 'Load jpg File');
x=[filename]

axes(handles.axes1)
rgb_img = imread(x); % Load the image
image(rgb_img) % Dis

axes(handles.axes2)
I = .2989*rgb_img(:,:,1)...
+.5870*rgb_img(:,:,2)...
+.1140*rgb_img(:,:,3);
min(I(:))
max(I(:))
imshow(I)

axes(handles.axes3)
BW1=edge(I,'Sobel');
imshow(BW1)
bwarea(BW1)
OriSize=bwarea(I)
Texture1=bwarea(BW1);
percentageA=100-((OriSize-Texture1)/OriSize)*100;
set(handles.edit1,'string',percentageA);
handles.myresult=percentageA;
guidata(hObject,handles)
```

```
% --- Executes on button press in pushbutton2.

function pushbutton2_Callback(hObject, eventdata, handles)
[filename, pathname, filterindex] = uiputfile( ...
{'*.m;*.fig;*.mat;*.mdl','MATLAB Files (*.m,*.fig,*.mat,*.mdl)';
'*.m', 'M-files (*.m)'; ...
'*.fig','Figures (*.fig)'; ...
'*.mat','MAT-files (*.mat)'; ...
'*.mdl','Models (*.mdl)'; ...
'*.*',  'All Files (*.*)'}, ...
'Save as');
```

```
% --- Executes on button press in pushbutton3.

function pushbutton3_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile('*.jpg', 'Load jpg File');
x=[filename]

axes(handles.axes4)
rgb_img = imread(x); % Load the image
image(rgb_img) % Dis

axes(handles.axes5)
I = .2989*rgb_img(:,:,1)...
+.5870*rgb_img(:,:,2)...
+.1140*rgb_img(:,:,3);
min(I(:))
max(I(:))
imshow(I)

axes(handles.axes6)
```

```
BW1=edge(I,'Sobel');
imshow(BW1)
bwarea(BW1)
OriSize=bwarea(I)
Texture1=bwarea(BW1);
percentageA=100-((OriSize-Texture1)/OriSize)*100;
set(handles.edit5,'string',percentageA);
handles.myresult2=percentageA;
guidata(hObject,handles)
```

```
% --- Executes on button press in pushbutton4.

function pushbutton4_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile('*.jpg', 'Load jpg File');
x=[filename]

axes(handles.axes7)
rgb_img = imread(x); % Load the image
image(rgb_img) % Dis

axes(handles.axes8)
I = .2989*rgb_img(:,:,1)...
+.5870*rgb_img(:,:,2)...
+.1140*rgb_img(:,:,3);
min(I(:))
max(I(:))
imshow(I)

axes(handles.axes9)
BW1=edge(I,'Sobel');
imshow(BW1)
bwarea(BW1)
```

```
OriSize=bwarea(I)
Texture1=bwarea(BW1);
percentageA=100-((OriSize-Texture1)/OriSize)*100;
set(handles.edit7,'string',percentageA);
handles.myresult3=percentageA;
guidata(hObject,handles)
```

```
% --- Executes on button press in pushbutton5.

function pushbutton5_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile('*.jpg', 'Load jpg File');
x=[filename]

axes(handles.axes10)
rgb_img = imread(x); % Load the image
image(rgb_img) % Dis

axes(handles.axes11)
I = .2989*rgb_img(:,:,1)...
+.5870*rgb_img(:,:,2)...
+.1140*rgb_img(:,:,3);
min(I(:))
max(I(:))
imshow(I)

axes(handles.axes12)
BW1=edge(I,'Sobel');
imshow(BW1)
bwarea(BW1)
OriSize=bwarea(I)
Texture1=bwarea(BW1);
percentageA=100-((OriSize-Texture1)/OriSize)*100;
```

```
set(handles.edit9,'string',percentageA);
handles.myresult4=percentageA;
guidata(hObject,handles)
```

```
% --- Executes on button press in pushbutton6.

function pushbutton6_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile('*.jpg', 'Load jpg File');
x=[filename]

axes(handles.axes13)
rgb_img = imread(x); % Load the image
image(rgb_img) % Dis

axes(handles.axes14)
I = .2989*rgb_img(:,:,1)...
+.5870*rgb_img(:,:,2)...
+.1140*rgb_img(:,:,3);
min(I(:))
max(I(:))
imshow(I)

axes(handles.axes15)
BW1=edge(I,'Sobel');
imshow(BW1)
bwarea(BW1)
OriSize=bwarea(I)
Texture1=bwarea(BW1);
percentageB=100-((OriSize-Texture1)/OriSize)*100;
set(handles.edit14,'string',percentageB);
percentageA=handles.myresult;
x=(percentageA+percentageB)/2;
```

```
no=0
moderate=1
medium=2
severe=3
bad=4
if x <= 0.555
set(handles.edit4,'string',no);
   elseif x <= 2 & x > 0.555
      set(handles.edit4,'string',moderate);
         elseif  x > 2 & x <= 3
            set(handles.edit4,'string',severe);
elseif x > 3 & x <= 4
         set(handles.edit4,'string',severe);
         else x > 4
          set(handles.edit4,'string',bad);
         end;
```

```
% --- Executes on button press in pushbutton7.

function pushbutton7_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile('*.jpg', 'Load jpg File');
x=[filename]

axes(handles.axes16)
rgb_img = imread(x); % Load the image
image(rgb_img) % Dis

axes(handles.axes17)
I = .2989*rgb_img(:,:,1)...
+.5870*rgb_img(:,:,2)...
+.1140*rgb_img(:,:,3);
min(I(:))
```

```
max(I(:))
imshow(I)

axes(handles.axes18)
BW1=edge(I,'Sobel');
imshow(BW1)
bwarea(BW1)
OriSize=bwarea(I)
Texture1=bwarea(BW1);
percentageB=100-((OriSize-Texture1)/OriSize)*100;
set(handles.edit15,'string',percentageB);
percentageA=handles.myresult2;
x=(percentageA+percentageB)/2;
no=0
moderate=1
medium=2
severe=3
bad=4
if x <= 0.555
set(handles.edit6,'string',no);
   elseif x <= 2 & x > 0.555
      set(handles.edit6,'string',moderate);
         elseif  x > 2 & x <= 3
            set(handles.edit6,'string',severe);
elseif x > 3 & x <= 4
         set(handles.edit6,'string',severe);
         else x > 4
          set(handles.edit6,'string',bad);
         end;
```

```
% --- Executes on button press in pushbutton8.
```

```
function pushbutton8_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile('*.jpg', 'Load jpg File');
x=[filename]

axes(handles.axes19)
rgb_img = imread(x); % Load the image
image(rgb_img) % Dis

axes(handles.axes20)
I = .2989*rgb_img(:,:,1)...
+.5870*rgb_img(:,:,2)...
+.1140*rgb_img(:,:,3);
min(I(:))
max(I(:))
imshow(I)


axes(handles.axes21)
BW1=edge(I,'Sobel');
imshow(BW1)
bwarea(BW1)
OriSize=bwarea(I)
Texture1=bwarea(BW1);
percentageB=100-((OriSize-Texture1)/OriSize)*100;
set(handles.edit16,'string',percentageB);
percentageA=handles.myresult3;
x=(percentageA+percentageB)/2;
set(handles.edit8,'string',x);
no=0
moderate=1
medium=2
severe=3
bad=4
```

```
if x <= 0.555
set(handles.edit8,'string',no);
   elseif x <= 2 & x > 0.555
      set(handles.edit4,'string',moderate);
         elseif  x > 2 & x <= 3
            set(handles.edit8,'string',severe);
elseif x > 3 & x <= 4
         set(handles.edit8,'string',severe);
         else x > 4
          set(handles.edit8,'string',bad);
         end;
```

```
% --- Executes on button press in pushbutton9.

function pushbutton9_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile('*.jpg', 'Load jpg File');
x=[filename]



axes(handles.axes22)
rgb_img = imread(x);
image(rgb_img)

axes(handles.axes23)
I = .2989*rgb_img(:,:,1)...
+.5870*rgb_img(:,:,2)...
+.1140*rgb_img(:,:,3);
min(I(:))
max(I(:))
imshow(I)
```

```
axes(handles.axes24)
BW1=edge(I,'Sobel');
imshow(BW1)
bwarea(BW1)
OriSize=bwarea(I)
Texture1=bwarea(BW1);
percentageB=100-((OriSize-Texture1)/OriSize)*100;
set(handles.edit17,'string',percentageB);
percentageA=handles.myresult4;
x=(percentageA+percentageB)/2;
set(handles.edit10,'string',x);
no=0
moderate=1
medium=2
severe=3
bad=4
if x <= 0.555
set(handles.edit10,'string',no);
  elseif x <= 2 & x > 0.555
    set(handles.edit10,'string',moderate);
      elseif  x > 2 & x <= 3
        set(handles.edit10,'string',severe);
elseif x > 3 & x <= 4
      set(handles.edit10,'string',severe);
      else x > 4
       set(handles.edit10,'string',bad);
      end;
```

```
% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
set(handles.edit17,'string',percentage);
rgb_img = imread(x); % Load the image
```

```
image(rgb_img) % Dis
I = .2989*rgb_img(:,:,1)...
+.5870*rgb_img(:,:,2)...
+.1140*rgb_img(:,:,3);
min(I(:))
max(I(:))
figure; colormap(gray(256)); image(I)
BW1=edge(I,'Sobel');
bwarea(BW1)
OriSize=bwarea(I)
Texture1=bwarea(BW1);
percentage=100-((OriSize-Texture1)/OriSize)*100;

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