DIGITAL IMAGE ULCER ASSESSMENT ON SMARTPHONES

By

MUHAMMAD FARIS BIN MOHAMAD SAUPE

FINAL PROJECT REPORT

Submitted to the Department of Electrical & Electronic Engineering
in Partial Fulfilment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronic Engineering)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

DIGITAL IMAGE ULCER ASSESSMENT ON SMARTPHONES

By

Muhammad Faris bin Mohamad Saupe

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

Approved:

_____

Prof. Ir. Dr. Ahmad Fadzil Bin Mohamad Hani

Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

SEPTEMBER 2013

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.

_____

MUHAMMAD FARIS BIN MOHAMAD SAUPE

ABSTRACT

Chronic ulcers can cause a severe pain to the patient. A digital image processing technique was currently used to monitor the healing progression of the ulcers patient in order to give a proper treatment. Red granulation tissue on an ulcer image is a healing indicator. Red granulation tissues contain a haemoglobin pigment reflecting the red colour of the tissue. Basically, digital image processing technique with the help of independent component analysis (ICA) is a process which it extracts the haemoglobin from the ulcer images and measured its area. The ability of ICA to separate independent sources would allow us to extract haemoglobin related pixels and measure the area haemoglobin. Currently, tools use for assessment is DSLR camera and computer. These two tools are lacks in portability and time consume. A novel approach is to carry out the digital image processing technique on the smartphone device to make it more portable and faster to carry out assessments. The author takes advantage of Android NDK (Native Development Kit) to build an application. OpenCV and UMF environment was used and both of them wrote in native code. Somehow, the research project could not be finish due to the time constrain. Good background knowledge of native language is an advantage for those who want to continue a research project. Author's believes that, by implementing digital image ulcer assessment on smartphone, it will provide a good comeback.

ACKNOWLEDGEMENT

First of all, I would like to express my greatest praises to Allah, God Almighty for all His blessing and guidance throughout my life, especially during this period for me to complete my Final Year Project I and II. For without Him, I am a mere nothing without any significance. Next, I would like to show my appreciation to Universiti Teknologi PETRONAS (UTP) for giving me the chance to do my Final Year Project as it have given me valuable amount of experience.

My sincerest thank to my supervisor, Prof. Ir. Dr. Ahmad Fadzil Bin Mohamad Hani for giving me the chance and trust to do this project under his supervision. The guidance that he has given me will always be a memory that I hold strong and cherish. Finally, I would express my sincerest apologies if in any time that I have done anything that is not to his liking. Thanks to the Final Year Project Coordinator, Dr. Nasreen Badruddin and the Final Year Project Committees for their information and guidance given for the project.

Thanks to all lecturers, technicians, and postgraduate students who had provided many valuable advices throughout the project. To both my parent, thank you for your deep understanding upon completing the project. Thank you also to my colleagues of Universiti Teknologi PETRONAS.

TABLE OF CONTENTS

CERTIFICATION OF APPROVAL ii
CERTIFICATION OF ORIGINALITY iii
ABSTRACT iv
ACKNOWLEDGEMENT v
CHAPTER 1: INTRODUCTION 1
1.1 Background of Study 1
1.2 Problem Statement 2
1.2.1 Ineffective used of traditional ulcer assessment 2
1.2.2 Inefficient in time use 3
1.2.3 Limited in term of portability and people usage 3
1.3 Objectives 4
1.3.1 To implement fast ICA, segmentation and pixel classification 4
1.3.2 To design an android based system for the ulcer assessment tool 4
1.4 Scope of study 4
CHAPTER 2: LITERATURE REVIEW 5
2.1 Ulcer: Treatments and Causes 5
2.2 Assessment on Ulcers 5
2.3 Digital Image Processing Technique 6
2.4 Principal Component Analysis and Independent Component Analysis 8
2.5 Clinical research area on mobile devices 11
2.6 Android Developer 13
2.7 Android Native Development Kit (NDK) 15
CHAPTER 3: METHODOLOGY 16
3.1 Research Methodology 16
3.1.1 First Stage: Digital Image Processing Technique 16
3.1.2 Second Stage: Android Application Development 20
3.2 Software and tools 24
3.3 Gantt Chart 25
3.3.1 First semester 25
3.3.2 Second Semester 26

3.3.3 Extension Period 27
3.4 Key Milestone 28
CHAPTER 4: RESULT AND DISCUSSION 29
4.1 Image Processing 29
4.1.1 Read Image and Size 29
4.1.2 Grayscale and Plot Histogram 31
4.1.3 Separate Red, Green and Blue component 33
4.2 Result on ulcer assessment 34
4.2.1 Color Correction 34
4.2.2 Granulation tissue detection 34
4.2.3 Segmentation of granulation tissue 36
4.3 Android Application 37
4.3.1 Working Mechanism 37
4.3.2 Button Interaction 40
4.3.3 Implement of Digital Image Processing 45
CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS 47
CHAPTER 6: REFERENCES 49
CHAPTER 7: APPENDICES 51
7.1 Appendix A : Main.java class 51
7.2 Appendix B : Start.java class 53
7.3 Appendix C: Some part of Android.mk source code 57

LIST OF FIGURES

Figure 1: Typical Stages of a Healing Ulcer* *Reproduced from Goldman & Salcido (2002) 6
Figure 2 : Absorption Spectra of Main Skin Pigments, Melanin, Deoxy-haemoglobin (Hb), Oxy-haemoglobin (HbO2), and Bilirubin* *Reproduced from R.R. Anderson and J. A. Parrish (1981) 7
Figure 3: Visible Light Reflectance Spectrum 7
Figure 4 : The original source 9
Figure 5 : The observed mixture of the source signal in figure 4 9
Figure 6 : The estimates of the original source signals, estimated using only the observed signal in Figure 5. The original signals were very accurately estimated, up to multiplicative signs. 10
Figure 7 : Whitening presenting (a) the joint distribution of the independent components s1 and s2 with uniform distribution, (b) the joint distribution of the observed mixtures x1 and x2, and (c) the joint distribution of the whitened mixtures z1 and z2 10
Figure 8 : The set of eigen images extracted from fMRI scan of brain tissue 11
Figure 9 : The basis vectors of fMRI scans gained by Fast ICA algorithm 11
Figure 10: Android System Architecture 13
Figure 11: Life Cycle of an Android Activity 14
Figure 12: JNI serves as a gateway between native code and Java 15
Figure 13: Timeline of the digital image processing technique 16
Figure 14: White sticker on patient leg 17
Figure 15: Flowchart of haemoglobin distribution detection 18
Figure 16: Basic step of android project application 20
Figure 17: Eclipse with ADT plugin and AVD 21
Figure 18: Basic Implementation of LoaderCallbackInterface 22
Figure 19: NDK Application Development Process 23
Figure 20: Project key milestone 28
Figure 21 : Image show from *imread* 30
Figure 22 : Grayscale image and its histogram 31
Figure 23: Black and white image with it histogram 32
Figure 24: Separated red, green, and blue channels 33
Figure 25: Histogram for red, green, and blue channels 33
Figure 26: Ulcer image before and after color correction 34
Figure 27 : Extracted independent sources from observed colour ulcer image 35
Figure 28: Classified Image obtained from haemoglobin image 36

Figure 29: Binary image obtained from classified image 36
Figure 30: Basic project android application working mechanism 37
Figure 31: Digital Image Ulcer Assessment interface 39
Figure 32: Intent dialog, choose either capture a picture or select from gallery 41
Figure 33: About dialog 44
Figure 34: UMF sources files located inside android application 46

LIST OF TABLES

Table 1 : Lists of software 24
Table 2 : Lists of hardware 24
Table 3 : Gantt Chart for Semester 1 25
Table 4: Gantt Chart for Second Semester 26
Table 5: Gantt Chart for Extension Period 27
Table 6 : Some of image/graphics format supported by imread and imwrite, starting with MATLAB 7.6 29
Table 7: Read image 30
Table 8: Grayscale and histogram plotting code 31
Table 9: *whos* source code 32
Table 10: Image to black white code 32
Table 11: Separate Red, Green and Blue component code 33
Table 12 : Source code for BaseLoaderCallback to initialise OpenCV 38
Table 13: Source code inside main.xml layout 39
Table 14: Crop image function code 42
Table 15: Source code to convert YUV into RGB code color 45

CHAPTER 1: INTRODUCTION

1.1 Background of Study

Leg ulcers refer to full thickness skin loss on the leg or foot due to any cause. There are two types of leg ulcers, which are acute ulcers and chronic ulcers. The one follows the normal phases of healing was called as acute ulcers; they were expected to show sign of healings in less than four weeks. For those that persists longer than four weeks were called having chronic ulcers [1]. Unlike acute ulcers, chronic ulcers may become worse as time travel.

Chronic ulcers usually found on the lower knee and affect around 1% of adult population. Chronic ulcers may cause severe pain to the patients if improper treatment was given. To avoid given an improper treatment to the patient, an ulcer assessment is needed to make sure that the treatment give a positive outcome and not a negative. Many of research have been done to find good tools for ulcers assessment and the newly found technique would be Digital Image Processing Technique. Digital image processing techniques provided are precisely, objectively, and reliable data compare to the traditional technique of ulcer assessment that is on simple visual inspection [2]. The growth of red granulation tissue – containing haemoglobin pigment – on the surface of ulcers is an indicator to the wound healing process. Digital image processing technique utilise this information by extracting the red granulation tissue from the image to monitor the healing process of the chronic ulcers. Independent Component Analysis (ICA) was used to extract the haemoglobin image from the ulcer image [3].

In current ulcer assessment, the tools used to capture the ulcer images was a DSLR camera. Then, the image will be loaded inside a computer to do an analysis; make assessment to the ulcer. The DSLR camera provided a high quality of image, but some of us may think that, the size of DSLR camera is big and difficult to carry out the task. For example, nursing staff has to carry around big DSLR camera just to take a picture of ulcer patient; ulcer image. Thus, the novel approach is to implement the digital image ulcer assessment on an android based smartphone. Rather than taking an ulcer image

using DSLR camera, an alternative way is to use a smartphone camera. Today, around 19.8 per cent of the world population owns a smartphone, which is about 1.4 billion active smartphone in use around the world. eMarketer reported that smartphone penetration worldwide will be double reaching 2.51 billion or 34 per cent by 2017 [4].

The implementation of digital image ulcer assessment on the smartphone will be provided a lot of benefit such as its can be used by anyone regardless of their area of background. For example, the parent can monitor children's wound. In 2009, there are more than half percent of physician that using a smartphone. By implementing digital image ulcer assessment on the smartphone, this will create an opportunity to expanding the area, in general medical area [5].

1.2 Problem Statement

1.2.1 Ineffective used of traditional ulcer assessment

There are some techniques used for ulcers assessment, such as simple visual inspection which is a popular method used on the clinical center. This technique lacks of precision and consistency, and it is very subjective. The simple visual inspection observed the ulcer to estimate the percentage area of the ulcer; granulation tissue percentage area. This technique really difficult because at one time, there are four tissues on the surface of ulcer; black necrotic, yellow slough, red granulation and pink epithelial. For example, it was difficult if you wanted to find the value of red colour on the colourful picture with consists of mixture of colour by only observing it. You can only guess it without knowing the exact amount of the red colour used. It is easy if you have only a red colour on the picture and it is a lot easy to estimate the amount of colour use. ICA can do better than this. Digital imaging technique used ICA because ICA have an ability to extract the independent source, granulation tissue from mixture sources, which is a colour images of ulcers. The segmentation and pixel classification is the additional principal used to calculate the percentage area of granulation tissue because ICA cannot do on its own.

1.2.2 Inefficient in time use

Although, digital image processing technique provided a fast result than traditional way, but it not yet been fully use. The current tool used – DSLR camera – is not really convenient in some cases. The processes of digital image processing technique required an ulcer picture which was captured by a camera (currently is DSLR camera), and then the images of ulcer need to undergo several processes before produced result. For example, the nurse needs to take a picture of a single patient that having ulcers to be analysed using a computer. The pictures were loaded into a computer to analyse and then the patient will know their result. But let's say there are 50 patients waiting for an assessment's, the nurse needed to take a picture of every patient and then go to computer to do analysis and then go back to every patient to give their result. The first patient waiting period is longer than other patients because waiting for the nurse to take picture of other patients. Using a smartphone compared to the DSLR camera and computer is really helpful in this case because the patients will get an instant result of the assessment after nurse took their ulcer picture with mobile device; smartphone in this case. The nurse's time that was used previously can be used for other importance task or work.

1.2.3 Limited in term of portability and people usage

1) Portability

Current digital image ulcer assessment tool is a DSLR camera and it is considered been big for some of us. This tool seems unreliable and weird when the nurse used it. Using a smartphone may solve this problem because smartphone is slim and it common thing for everyone.

2) People usage

For the timing, digital image ulcer assessment can only be used in the healthcare center. This means everyone that need to make an assessment on how their healing processes of their ulcers need to go to healthcare center regardless of their condition, either the normal wound or the chronic ulcers. Digital image ulcer assessment on smartphone not only can be used by healthcare center but to other people regardless of their background.

1.3 Objectives

1.3.1 To implement fast ICA, segmentation and pixel classification

This project is evaluated the healing process of ulcer. The fast ICA need to be performed in order to extract the haemoglobin related pixels from the ulcer digital image. This pixel is a measurement to the healing process. Segmentation and pixel classification need to be done in order to measure the area of the granulation tissue which will be indicated the healing process of the ulcer.

1.3.2 To design an android based system for the ulcer assessment tool

Current tool used to take an ulcer images is DSLR camera, as we know that it's big and heavy. The tool is not reliable in term of portability. Furthermore, people tend to carry something, which is lighter rather than heavy. So the smartphone android application was created in order to solve the portability part. Smartphone is very popular among us because it is lighter and portable. When there is smartphone application for ulcer assessment, so there are no more DSLR camera to carry out the task; to made an ulcer assessment to patient. The smartphone is very reliable and it is easy to carry around.

1.4 Scope of study

Some scope of study is as below

1) To understand the chronic ulcers, the cause, effect, assessment and treatment

2) To learn on evaluating the healing process of chronic ulcers using digital image processing technique

3) To learn image processing in Matlab

4) To learn about the android programming language

5) To create an android based application

CHAPTER 2: LITERATURE REVIEW

2.1 Ulcer: Treatments and Causes

Ulcers are chronic wounds that fail to heal [2]. There are many types of chronic ulcers, some of them are venous, arterial, diabetic, pressure and inflammatory ulcers [6]. The common part of our body that mostly having an ulcer are lower extremity below the knee and can also be called as leg ulcer [7]. Leg ulcer was referred to full thickness skin loss on the leg or foot due to any causes. Chronic leg ulcer commonly occurred after a minor injury in association with chronic venous insufficiency, chronic arterial insufficiency, diabetes and hypertension. Most of the patience suffered severe pain because of the non-healing ulcer. There are many types of treatment based on the type of ulcers. These treatments are to reverse the factors that have caused the ulcers. If the patients have venous leg ulcers only without arterial disease, usually they will be treated with exercise, elevation at rest, and compression. Compression cannot be used to the patient with arterial disease because it will aggravate an inadequate blood supply. Surgery also can help if the ulcer is deep venous system. Common treatment to take care of the skin was given to all kinds of ulcers. Careful assessment need to be done in order to give a right treatment to the patient [8].

2.2 Assessment on Ulcers

The ulcer colour changes from black to yellow to red gradually as it heals. The each ulcer colour represents tissues on the ulcer surface. There are four tissues in the ulcer surface, which are black necrosis, yellow slough, red granulation and pink epithelial as shown in figure 1. These tissues will appear on the ulcer surface as they progress throughout the healing process [2, 9]. These tissues appear to be important information for the healing status of the wound. The black necrotic tissue mostly appears first and followed by red granulation tissue started to grow, then yellow slough and lastly pink epithelial tissue that eventually close the ulcers.

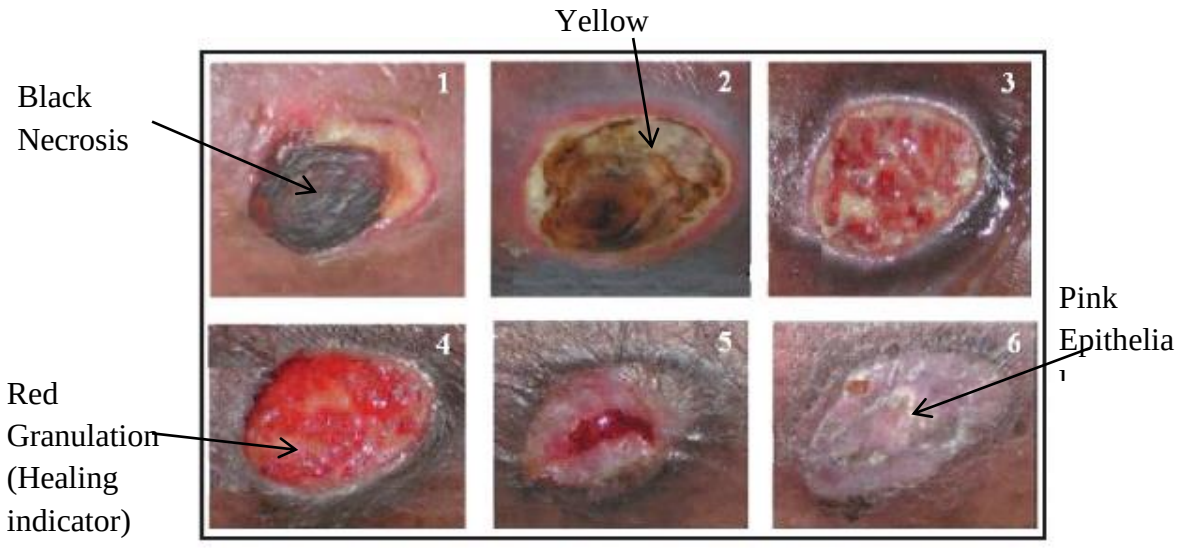


Figure 1: Typical Stages of a Healing Ulcer*
*Reproduced from Goldman & Salcido (2002)

Clinical method to monitor the healing progress of the wound is by measured the amount of each tissue based on the visual inspection of the wound. At any one time all four tissue types can be present on the ulcer surface and it is subjective, lack precision and consistency made this method more difficult and challenged. Today, there is a method that provided more precise, objective and reliable data called as a digital image processing technique. Although it a new discovery, but it is the effective way to make the assessment on wound; chronic ulcers. The digital image processing technique is based on the haemoglobin content in chronic ulcers as an image marker, to detect of chronic ulcers [3].

2.3 Digital Image Processing Technique

Digital Image Processing Technique used colour images of chronic wounds to measure the healing process. Three main approached in this area of study are information from a single colour channel, RGB histogram distributions and colour and texture descriptors. Single colour channel is an early developed method for wound tissue assessment based colour information by using conventional colour model RGB and HIS. This method was developed by Herbin *et al.* by digitized RGB images of artificial created wounds [3, 9].

While colour and texture descriptor method is the most recently method has been developed on ESCALE project. This unsupervised wound tissue segmentation method was proposed to design a complete 3D and colour wound assessment tool.

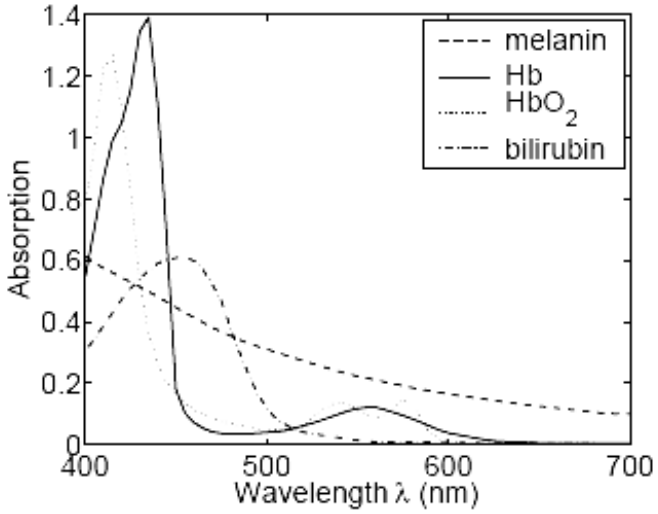


Figure 2 : Absorption Spectra of Main Skin Pigments, Melanin, Deoxy-haemoglobin (Hb), Oxy-haemoglobin (HbO2), and Bilirubin*
*Reproduced from R.R. Anderson and J. A. Parrish (1981)

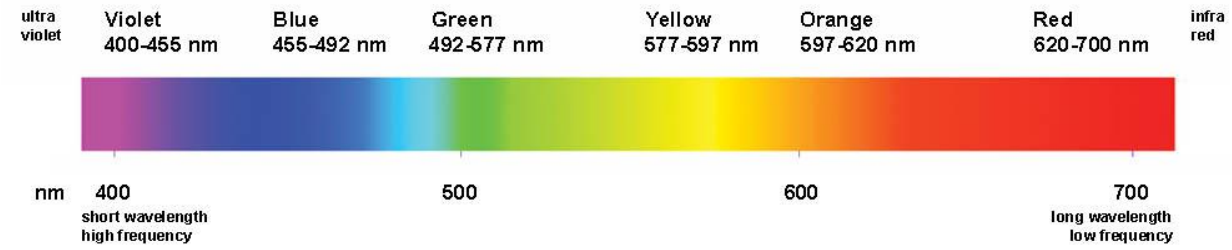


Figure 3: Visible Light Reflectance Spectrum

Studies show that haemoglobin contains certain optical character that can be detected from coloured images and used to show their content within human skin. Figure 2 shows the absorption spectra of Main Skin Pigments, Melanin, Deoxy-haemoglobin (Hb), Oxy-haemoglobin (HbO2), and Bilirubin. Pigment hemoglobin exhibited its high absorbance at wavelengths around 420-430 nm and with no absorbance (total reflectance) for wavelengths above 640nm. The portion of the electromagnetic spectrum that is visible to

the human eye extended from violet about 380 nm to red which is about 750 nm as shows in figure 3. Red colour component existed at wavelength between 620-700 nm from the visible light. This explains why the blood that contains haemoglobin pigment appeared red in colour when viewed under visible light.

Red granulation tissue is appeared on the ulcer surface indicate that the ulcer is undergone a healing process and it red because of the haemoglobin. Oxy-haemoglobin and deoxy-haemoglobin was reported by Anderson et al. as absorption spectra of haemoglobin. The technique was used to extract the spectrum of haemoglobin is Independent Component Analysis (ICA). "Independent component analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals"[10]. According to its meaning, ICA able to separate independent signal sources, in this case to extract haemoglobin related pixels; represented granulation tissue. The healing progression is indicated by the amount of these pixels.

2.4 Principal Component Analysis and Independent Component Analysis

Both Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are transformations that rely on statistics of the given data set. The result obtained through ICA is presumed to be more meaningful than the one gained by PCA because PCA is based on the information given by second order statistics, whereas ICA goes up to high order statistics. But it better if ICA worked on the data that has been preprocessed by PCA [11].

Let used a following problem for the better understanding of how ICA works. Imagine there are two people talking simultaneously and the conversation was recorded by using two microphones in different locations. Let us their speech as sources *s* and the mixture of speech from recorded speech as *x*. The mathematical representation can be seen as

$$x_1(t) = a_{11}s_1 + a_{12}s_2 \quad (1)$$

$$x_2(t) = a_{21}s_1 + a_{22}s_2 \quad (2)$$

Where the elements a_{11}, a_{12}, a_{21}, and a_{22} are unknown mixing factors that depend on the distances of the microphones from the speakers.

The task is to find the original sources with no prior information about the mixing matrix A. This problem called as *cocktail-party problem*. ICA is a transformation that enables us to find such sources.

Figure 4 illustrated the original source signal 1 and 2, meanwhile figure 5 shows the observed mixtures of the source signal in figure 4. Figure 6 shows the separated mixture signal into two independent components

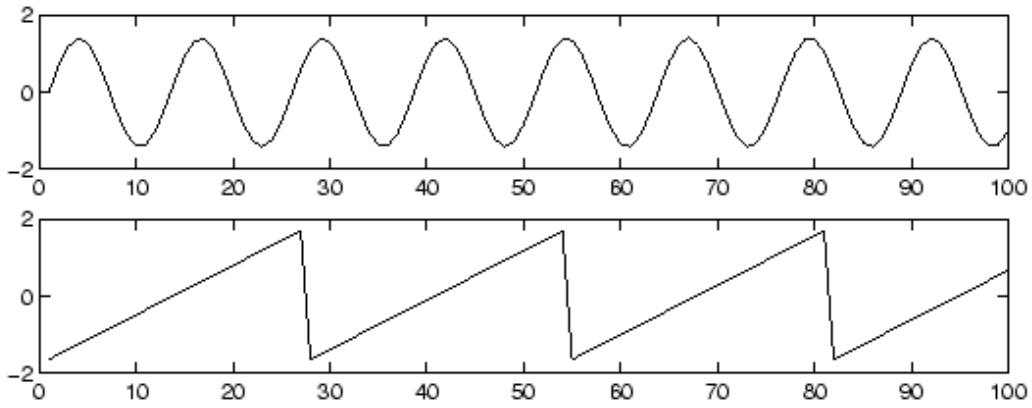


Figure 4 : The original source

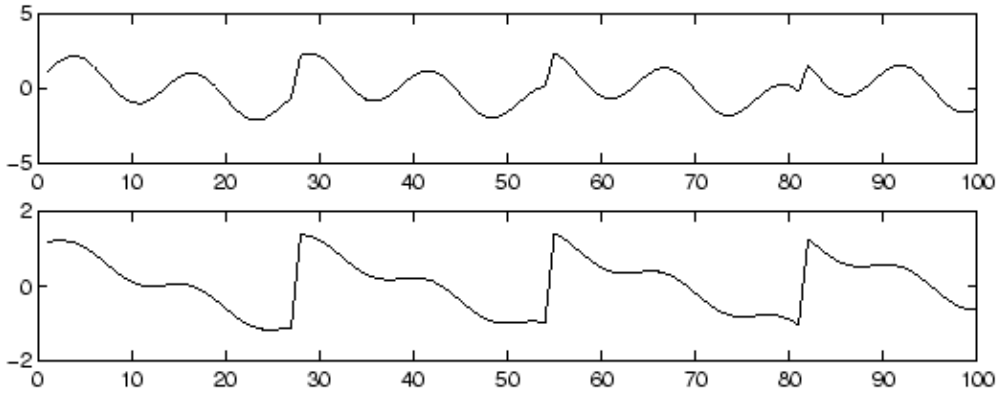


Figure 5 : The observed mixture of the source signal in figure 4

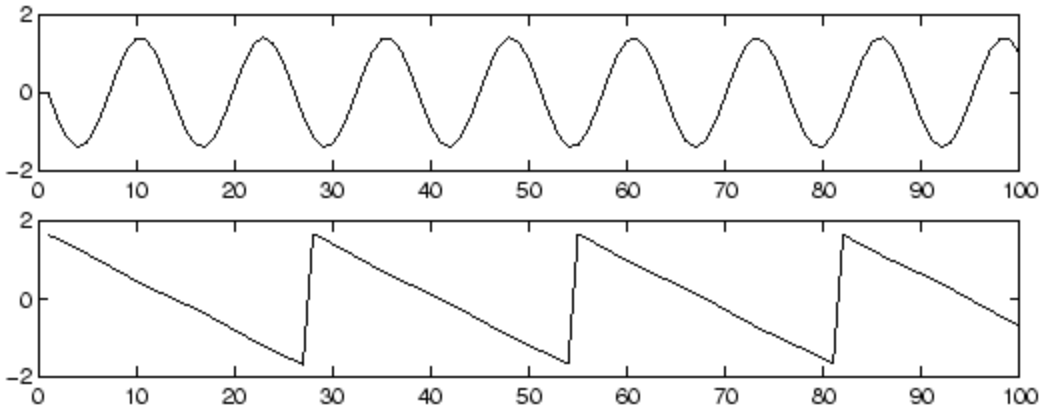


Figure 6 : The estimates of the original source signals, estimated using only the observed signal in Figure 5. The original signals were very accurately estimated, up to multiplicative signs.

Whitening often has been employed when estimating ICA model. Let's take an example of random vector z that possess following property

$$E\{z_i z_j\} = \delta_{ij}$$

On the equation above, it could be said that the vector **z** to be uncorrelated and of unit variance. The above mentioned properties is called white (sphered) for any vector z. Whitening (sphering) still did not given us independent component but it is still very important step when preprocessing the data (PCA). The last step when performing ICA is choice of the matrix that is already assumed to be orthogonal.

Figure 7 shows the whitening on source 1 and 2 for 3 situations.

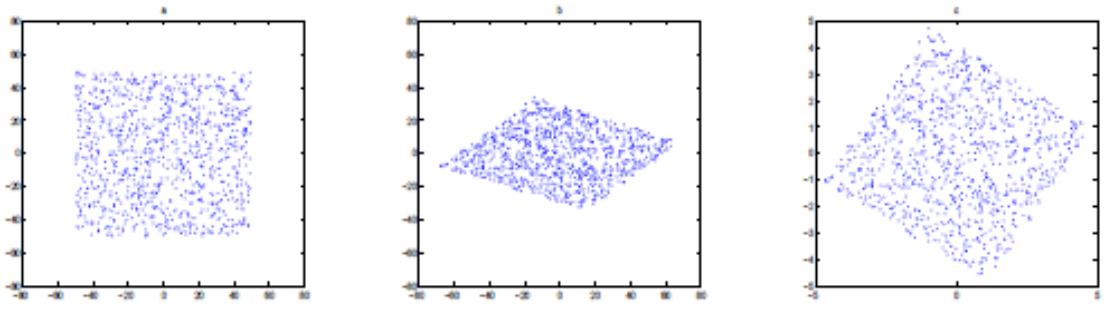


Figure 7 : Whitening presenting (a) the joint distribution of the independent components s1 and s2 with uniform distribution, (b) the joint distribution of the observed mixtures x1 and x2, and (c) the joint distribution of the whitened mixtures z1 and z2

The main concept of ICA applied to images insists on the idea that each image (subimage) may be perceived as linear superposition of features $a_i(x, y)$ weighed by coefficients s_i.In case of ICA, features are represented by columns of mixing matrix A and s_i are the elements of appropriate sources. In addition, ICA features are localised and oriented, and sensitive to lines and edges of varying thickness of image shown on figure 7 and 8. Furthermore, the sparsity of ICA coefficients should be pointed out. It is expected that suitable soft-thresholding on the ICA coefficients lead to efficient reduced of Gaussian noise [11, 12].

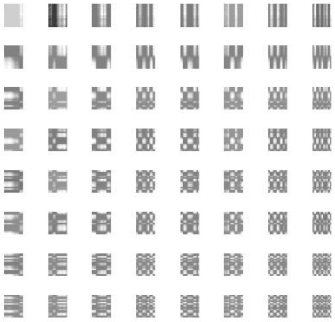


Figure 8 : The set of eigen images extracted from fMRI scan of brain

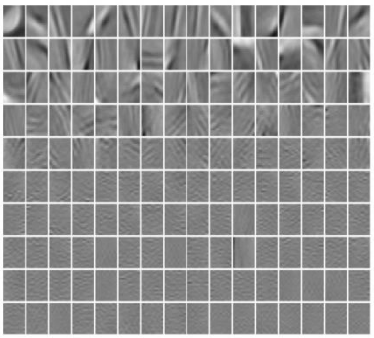


Figure 9 : The basis vectors of fMRI scans gained by Fast ICA algorithm

2.5 Clinical research area on mobile devices

The image normally will be taken by Digital Single Lens Reflector (DSLR) camera. In this study, the image will be taken by using a smartphone camera; Android Operation System (OS). There are some disadvantages taken an image by using a DSLR camera. One of disadvantages is a time-consuming. It took time to take the ulcer images, then analysed by using a computer. The clinical approach using a smartphone is not new, Godwin et al. has developed novel smart device based application for serial wound imaging. They created a BurnBookApp that is an innovative smart device application providing user-friendly serial imaging and informatics capabilities at the patient bedside to addressing the limitations of traditional photography and meeting the needs of burn clinicians. They reported consistent wound imaging and informatics are both feasible on a smart device platform [13].

Morak et al. in their research shown that the mobile based application is easy to use, intuitive and time-saving data acquisition without rely on particular infrastructure being present within a clinical environment. Recently investigational medical device (IMD) in the market is CardioMon (Medifina Medizinprodukte-Vertriebs GmbH, Vienna, Austria), which is a mobile medical device to measure a range parameters of the cardiovascular system. Morak et al. suggest that IMD should be embedded in an electronic data capture (EDC) system. This will enable the mobile devices to collect and document all measurement as well as additional information to perform further analyses [14].

Those research have proven that mobile devices, or in our case a smartphone have high opportunities to enhance in clinical research area. Based on the Pew Internet Project's research, 56% of American adults have a smartphone which has been reported in May 2013 [15]. Most of the users used a smartphone to take a picture. In 2009, there were 64% of physician used a smartphone application and is expected to grow as the time goes by [5]. Android operating system was chosen as the platform because IDC Worldwide Mobile Phone Tracker reports that in 2013, android operating system shared a largest smartphone operating system in the market worldwide [16]. A paper from Postolache et al. presented the design and implementation of a mobile TeleCare system using an Android OS smartphone, which is a system for long-term monitoring of vital signs and daily activities in patients [17]. Furthermore, in market nowadays, there are applications that use camera to detect a pulse by analysing the colour of the skin using a camera in android operating system such as Instant Heart Rate [18]. In addition, the Tianyu et al. demonstrates the capability of android smart phone camera used for pulse signal monitoring [19]. The system based on smartphone has good practical value for people who need convenient mobile medical service because of it portability and easy usage [13, 19].

2.6 Android Developer

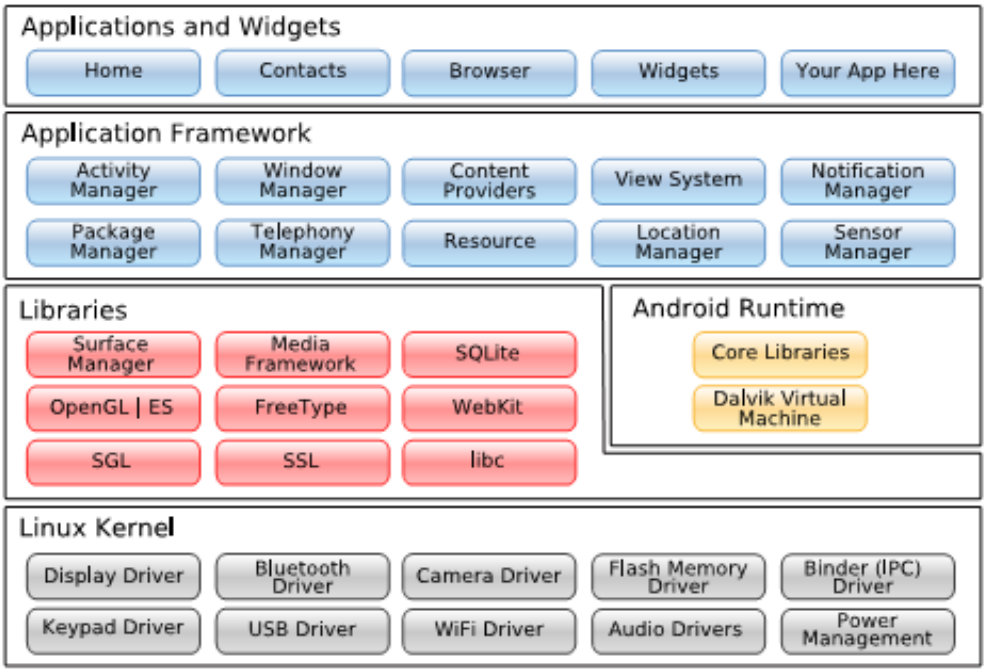


Figure 10: Android System Architecture

Figure 10 shows the android system architecture. Android is built on top of a solid and proven foundation: the Linux kernel. The program cannot call the Linux directly instead there have a helper, which is Dalvik Virtual Machine (VM). The next layer above the kernel contains the Android native libraries. Dalvik virtual machine is located on the Android Runtime, sitting also on the above the kernel layer. One of the Google's implementation of Java is on Dalvik VM, which was optimized for mobile devices. All code will be written in Java and run within the VM. Above the libraries and runtime layer, there is Application Framework layer. It provides the high-level building blocks and the topmost layer is Application and Widget layer. End users can only see the topmost layer. [20, 21]

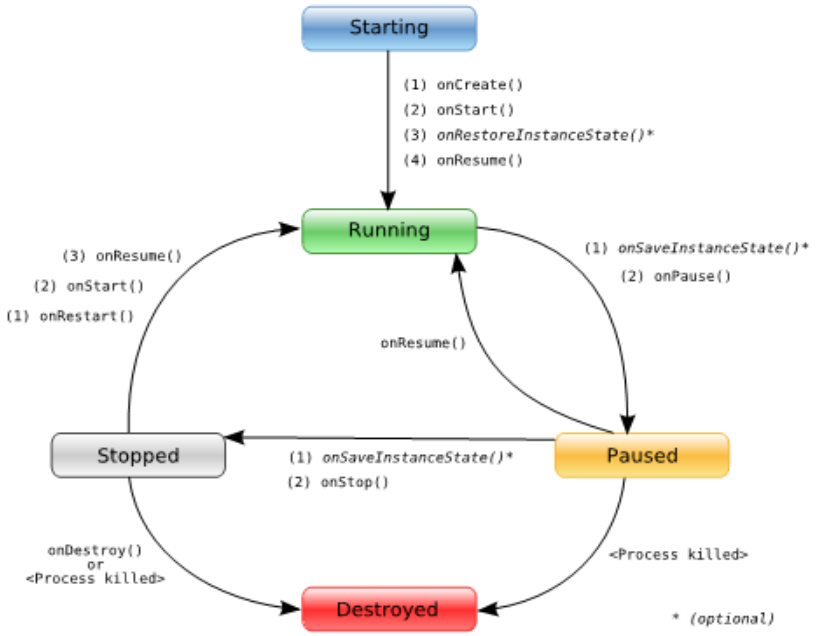


Figure 11: Life Cycle of an Android Activity

Android is different from Linux or Window which you can have many applications running and visible at once in different windows. Android cannot run and visible many applications at once. Android only has one foreground application, which typically takes over the whole display except for the status line. Each user interface screen is represented by an *Activity* class and each activity has its own life cycle. Figure 11 shows the example of activity life cycle. [20]

Android SDK defined a few objects that need to know for every developer to be familiar with. The most important ones are listed below:

1) An *activity* is a user interface. Applications can define one or more activities to handle different phases of the program.

2) *Intent* is a mechanism for describing a specific action such as “phone home”.

3) A *service* is a task that runs in the background without the user’s direct interaction.

4) A *content provider* is a set of data wrapped up in a custom API to read and write it.

2.7 Android Native Development Kit (NDK)

The NDK is a toolset that allows us to implement parts of our app using native-code languages such as C and C++. The NDK allows us to use part of our applications using native code languages such as C and C++[22].

There is advantage of using NDK because it's build for performance. But in the meantime, it increases application complexity. The performance improvement can come from three sources. Firstly, the native code is compiled to a binary code and run directly on OS, while Java code is translated into Java byte-code and interpreted by Dalvik VM. The developers was allowed to make used of some processor features that are not accessible at Android SDK, such as NEON, a Single Instruction Multiple Data (SIMD) technology, allowing multiple data elements to be processed in parallel would be the second sources performance improvement. The third aspect is that the developer can optimise the critical code at an assembly level, which is a common practice in desktop software development[23].

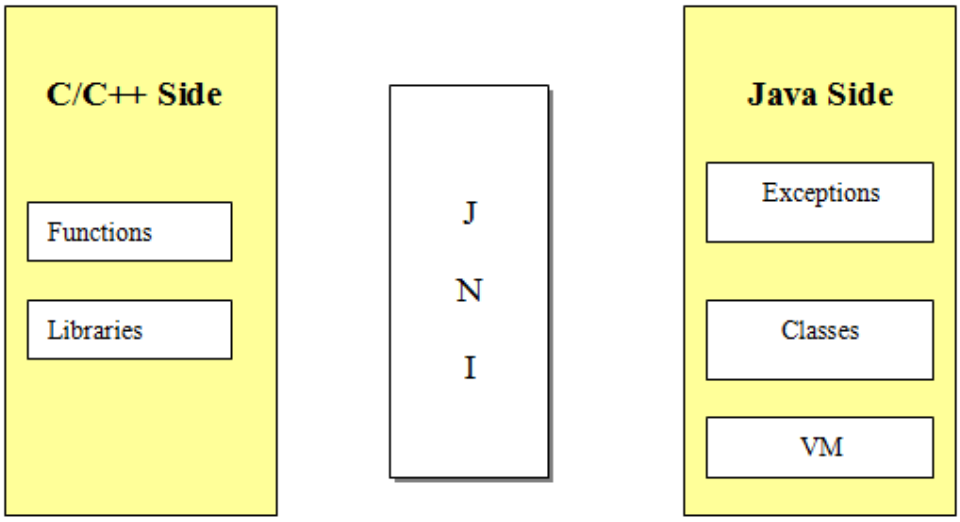


Figure 12: JNI serves as a gateway between native code and Java

In order for the native source code to work with android application, the JNI (Java Native Interface) is acting like a gateway to connect between Java and Native code. With JNI, Java applications that use the JNI can incorporate native code written in language such as C, C++, and Assembles, as well as code written in the Java programming language[24].

CHAPTER 3: METHODOLOGY

3.1 Research Methodology

On the research methodology part, there are two main stages. Firstly, on understanding the digital image processing technique and its algorithm, and secondly to create an android based application and implementation of digital image processing technique.

3.1.1 First Stage: Digital Image Processing Technique

During this stage, there are lots to learn about digital image processing technique. Three more stages was divided according to timeline of the digital image processing technique, which are before, during and after the digital image processing technique take place.

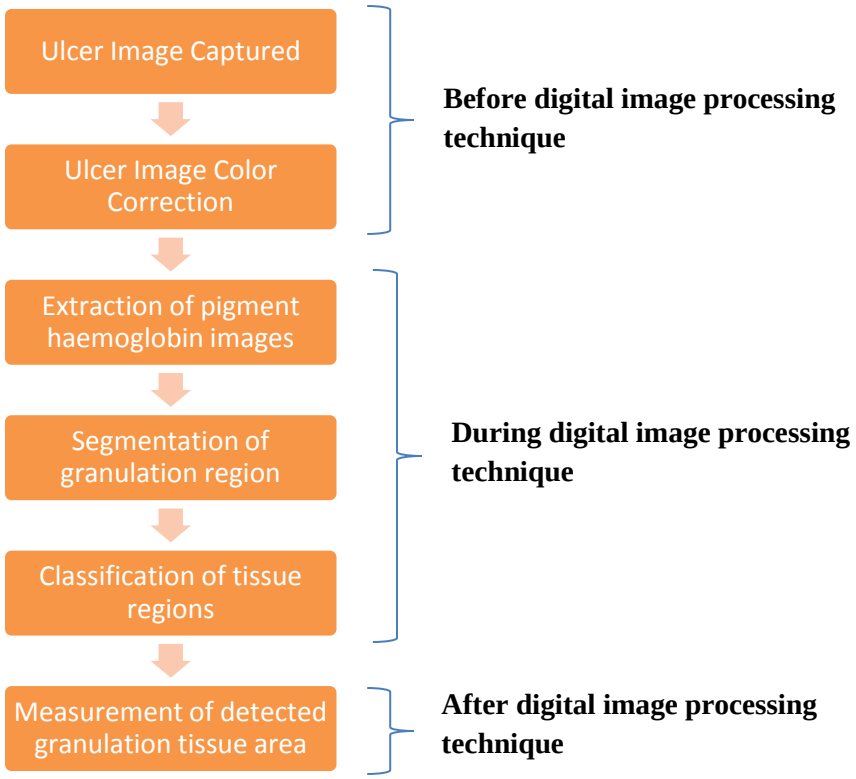


Figure 13: Timeline of the digital image processing technique

Before the process of digital image processing technique

The early task needs to be done first before starting the ulcer assessment. There are precautions need to be taken before starting to take the picture of ulcer on the patient. First thing first is on the patient that having an ulcer. The color sticker was stick on near the ulcer area on the patient before taking the ulcer image as shown on figure 14. The color of sticker chosen must be contra or not similar to the skin color. Basically, an ulcers images taken was affected by the surrounding, such as light intensity and the length of the ulcer area to the camera lens. The data will be less accurate if the problem does not solve. It's difficult to fix the length and also the light intensity on the room, so as a solution to the problem, a color sticker used. The color sticker play important roles to synchronize the images pixels into same light intensity which it's simply call as color correction. Color sticker also used as a reference for the pixel sizing later on. The images of ulcers were taken and saved for the next process.

Figure 14: White sticker on patient leg

During the process of digital image processing technique

The next step would be applied the digital image processing technique to the ulcers images. Basics steps of how to detect the haemoglobin distribution as figure 15.

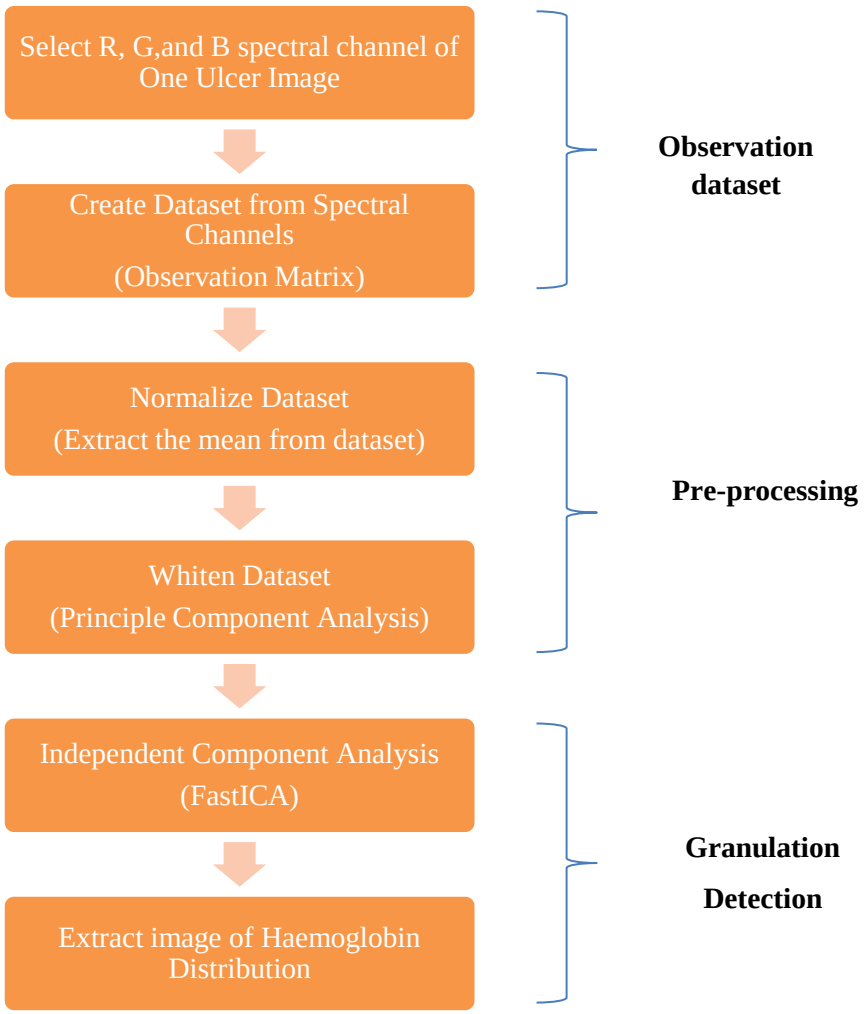


Figure 15: Flowchart of haemoglobin distribution detection

Based on figure 15, the observation dataset can be seen on the first two step of the flowchart. The RGB spectral channels were separated from the ulcer image. The spectral channel was stored on the dataset or observation matrix. Then, move on to next two steps will be pre-processing the dataset. The dataset is first normalised to centre on zero point by extracting the mean value from each spectral band. After the dataset has been normalizing, principal component analysis was used to whiten the dataset. The major process is the next step, which is Independent Component Analysis or Fast ICA. FastICA

play important part which it enables the separation of the entire element on the ulcer image. The user is needed to choose back which element is represented the haemoglobin part by their knowledge of it.

Then segmentation is implemented on the detected granulation tissue to segment the areas and make it more meaningful. Clustering is a common technique of unsupervised learning and classification where elements of a dataset are partitioned into several disjoint groups (cluster) so the points in one group are similar to each other and are as different as possible from points in other groups. In this case, K-mean clustering was used to classify the extracted haemoglobin images into distinctive cluster based on the Euclidean distance of each pixels value to the mean of the cluster centre. Classified image is then converted into binary image to measure the pixels.

After the process of digital image processing technique

After the image processing technique, the size of haemoglobin distribution of the individual image was stored and the graph of healing area against the duration of time was plotted.

3.1.2 Second Stage: Android Application Development

This stage is to implement the digital image processing technique on the android based application.

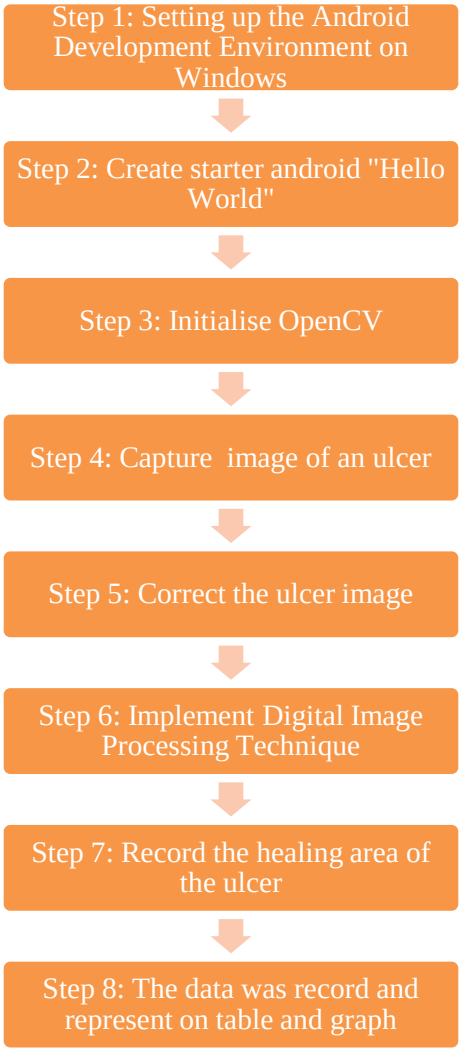


Figure 16: Basic step of android project application

Basically, the process of android application development can be seen in figure 16.

First step would be setup the Android Development Environment on Windows. There are some tools that need to be installed in order to complete the android development environment

1) Java Development Kit

2) Eclipse IDE

3) Install the Android SDK starter package

4) In Eclipse, install the ADT (ADT Developer Tools) plugin.

The starter android application “Hello world” was created. This is an empty project [25]. It’s shown the “Hello world” sentence on the opening of the application. The Android Virtual Device (AVD) was created in Android Virtual Device Manager (AVD Manager). Basically, the AVDs have a same function as a normal android smartphone except for the certain hardware for example camera, it did not include unless it is configured to webcam.

Figure 17: Eclipse with ADT plugin and AVD

OpenCV manager was initialise before android application start any activity. OpenCV (Open Source Computer Vision Library) is an open-source-BSD licensed library that includes several hundreds of computer vision algorithms written in C/C++. OpenCV designed for computational efficiency and with a strong focus on real-time applications. There are several modules on OpenCV, such as *core*, *imgproc* and *highgui*. *Core* modules is a compact module defining basic data structures, *imgproc* modules is an image processing module and *highgui* module is an easy-to-use interface to video capturing, image and video codecs, as well as simple UI capabilities. Figure 18 is basic implementation of LoaderCallbackInterface to initialise OpenCV.

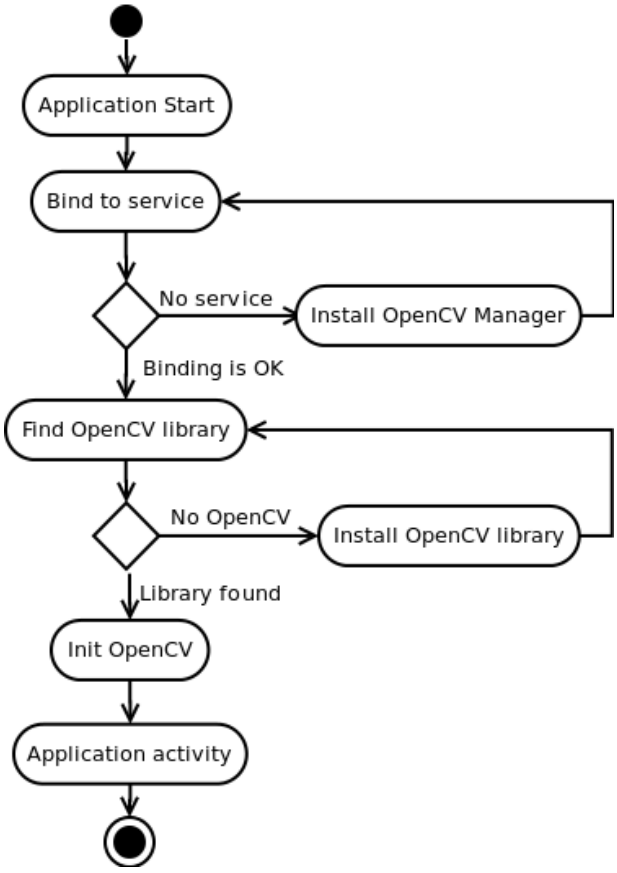


Figure 18: Basic Implementation of LoaderCallbackInterface

On step four, the image of the ulcer was captured. This image was taken by using the android based smartphone camera. The image will be stored inside the SD card of the smartphone and ready for analysis. Then the color correction to the ulcer image was done

by cropping the sticker that previously stick near the ulcer surface. This sticker will be a referenced to the ulcer image color correction.

Digital image processing technique implemented on the corrected image to extract the haemoglobin size. Fast ICA algorithm method used based on UMF environment. UTP-Matlab like Framework (UMF) is an environment to ease the problem of the conversion from MATLAB source code to C/C++ source code in Android OS. This framework is based on IT++, LAPACK and BLAS library, which area mathematic and signal processing libraries. The user can easily use matrix calculation and MATLAB-like notation when installing the framework in the application. The JNI help UMF environment to interact each other with Java. Figure 19 shows process flow of native source code until the execution of an application.

Figure 19: NDK Application Development Process

After the Digital image processing technique has been done, the android application would save the data of the haemoglobin size as healing area. All the saved data was represented on the graph of healing area against the duration of time.

3.2 Software and tools

There are several software tools and hardware tools have been use throughout the project. Some of important software and hardware are:

Table 1 : Lists of software

Software	**Description**
Matlab	To execute the digital image processing technique
ECLIPSE With ADT plugin	To create, debug and compile an android based application
Android Virtual Device	To test virtually the android application

Table 2 : Lists of hardware

Hardware	**Specification**	**Description**
Android Smartphone	Android Version 4.1.1 with camera	To test and debug the android application

3.3 Gantt Chart

3.3.1 First semester

Table 3 : Gantt Chart for Semester 1

		First Semester													
No	**Detail / Week**	**1**	**2**	**3**	**4**	**5**	**6**	**7**		**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	***Selection of Project Topic***															
2	***Understand the research*** *- Understanding the ulcer, cause and treatment* *- Research on Ulcer assessment* *- Extract RGB from digital image*							*1								
3	***Understanding PCA and its algorithm*** *- Whiten Data Set*								**Midsem Break**							
4	***Understanding the concept of ICA***															*2
5	***Understanding the algorithm of ICA***															
6	***Understanding segmentation***															
7	***Learn Basic Android tools*** - *Familiar with Android SDK*															

*1 Able to do a basic image processing on Matlab

*2 Know how the ICA work

3.3.2 Second Semester

Table 4: Gantt Chart for Second Semester

		Second Semester														
No	**Detail / Week**	**1**	**2**	**3**	**4**	**5**	**6**	**7**		**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	***Cont.- Understanding ICA algorithm***																
2	***Learning Java for Android programming language***																
3	***Create basic android application***			*3													
4	***Create GUI for the project***																
5	***Create pre-processing android application*** - *Capture and capture image* - *Implement PCA*																
6	***Implement ICA algorithm to android platform*** - *Writing an ICA algorithm*							*4									
7	***Implement segmentation to android platform*** - *Writing segmentation and classification algorithm*																
8	***Testing and debugging***																
9	***Verification and validation***																
10	***Writing a Report***																
11	***Final Report Submission***																*5

Midsem Break

*3 Able to create Android Application

*4 Draft how to implement ICA on Android

*5 Able to implement Digital Image Processing Technique on android application

3.3.3 Extension Period

Table 5: Gantt Chart for Extension Period

		Extension Period			
No	**Detail/Week**	**1**	**2**	**3**	**4**
1	***Learning Native Android***				
2	***Implementation of ICA***				
3	***Test and Debug***				
4	***Report and Presentation***				

3.4 Key Milestone

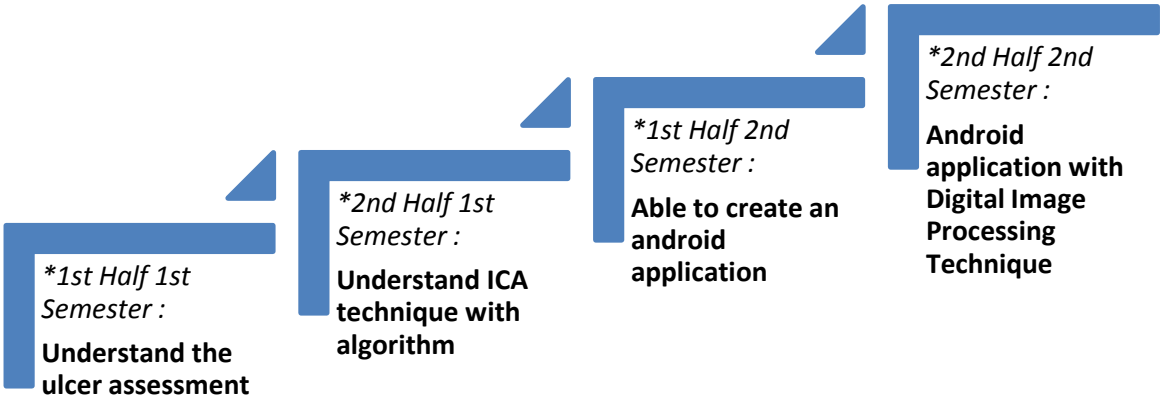


Figure 20: Project key milestone

Key milestone of project was dividing into 4 timelines. On first half of first semester study, the understandings of the ulcer assessment have been done on that timeline. During second half of first semester study, the learning of ICA and its algorithms is started. Then, move to second semester study. The first half of second semester study is to learn android application and the end of the timeline is able to create a basic android application. Lastly, on second half of second semester, it starts with creating and android interface to implement digital image processing technique. At the end of the timeline, the digital image ulcer assessment on android application should be finished.

CHAPTER 4: RESULT AND DISCUSSION

4.1 Image Processing

Some basic image processing of have been learn such as read image from graphic file, make a gray scale image, plot a histogram from a gray scale image, separate red, green and blue Component. Matlab was used to execute the entire task.

4.1.1 Read Image and Size

Matlab code function *a=imread(filename)* used to read image from graphic file.

"a" is an array of image file load, while the *filename* is the location of the image file inside the computer. The image format was included inside the filename. Table 6 shows the image formats supports by *imread.*

Table 6 : Some of image/graphics format supported by imread and imwrite, starting with MATLAB 7.6.

Format Name	Description	Recognized Extensions
BMP†	Windows Bitmap	`.bmp`
CUR	Windows Cursor Resources	`.cur`
FITS†	Flexible Image Transport System	`.fts, .fits`
GIF	Graphics Interchange Format	`.gif`
HDF	Hierarchical Data Format	`.hdf`
ICO†	Windows Icon Resources	`.ico`
JPEG	Joint Photographic Experts Group	`.jpg, .jpeg`
JPEG 2000†	Joint Photographic Experts Group	`.jp2, .jpf, .jpx, j2c, j2k`
PBM	Portable Bitmap	`.pbm`
PGM	Portable Graymap	`.pgm`
PNG	Portable Network Graphics	`.png`
PNM	Portable Any Map	`.pnm`
RAS	Sun Raster	`.ras`
TIFF	Tagged Image File Format	`.tif, .tiff`
XWD	X Window Dump	`.xwd`

†Supported by `imread`, but not by `imwrite`

The size of picture can be by seen by typing the function *size(a)*. The size will be 538x800 pixels in 3 dimensions color. *Imshow(a)* will showing the image of inside the array "*a"*. Figure 21 shows the result of the code below.

Table 7: Read image

```
>> a = imread('C:\Users\farisms\Pictures\Bachalpseeflowers.jpg');
>> size(a)

ans =

  538   800     3

>> imshow(a)
```

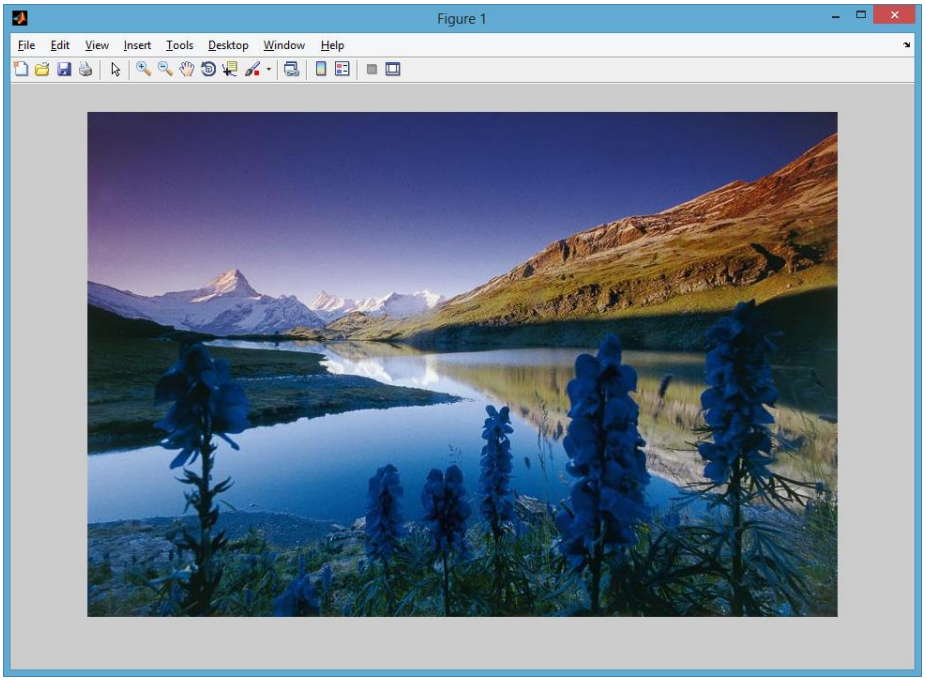


Figure 21 : Image show from *imread*

4.1.2 Grayscale and Plot Histogram

To convert the color image into the grayscale, the function *rgb2gray* was used. As the syntax call, it's easy to tell what exactly that the function does. The function *imhist* was showing the histogram chart of the grayscale picture.

Table 8: Grayscale and histogram plotting code

```
>> a = imread('C:\Users\farisms\Pictures\Bachalpseeflowers.jpg');
>> b = rgb2gray(a);
>> subplot(2,1,1), imshow(b), title('Grayscale image')
>> subplot(2,1,2), imhist(b), title('Histogram of grayscale image')
```

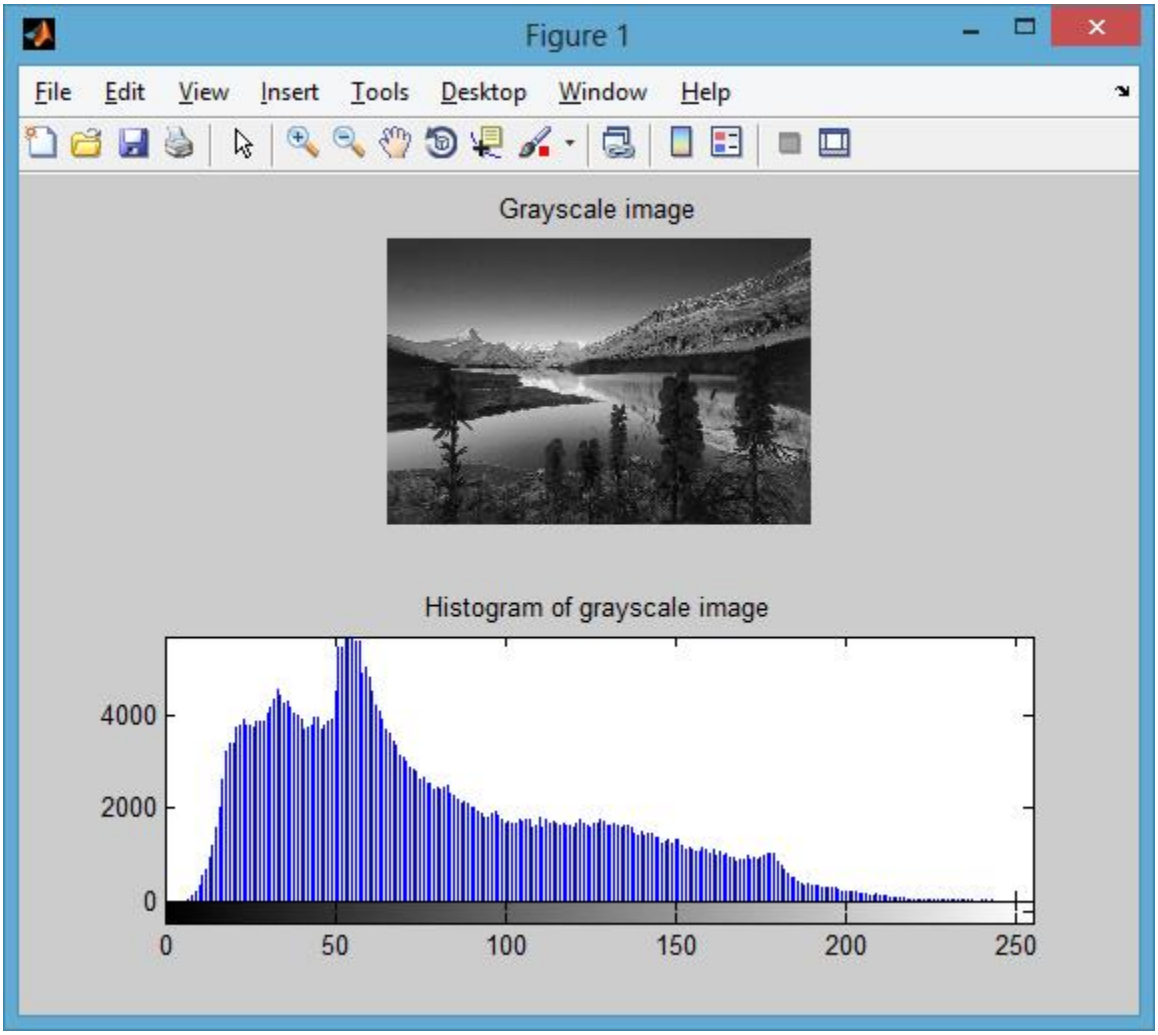


Figure 22 : Grayscale image and its histogram

Figure 22 shows the picture of grayscale image and histogram of the grayscale image

Histogram shows the pixels on the specific light intensity. The function *imhist* uses a default value of 256 bins because of a grayscale image have an integer value in the range [0,255] for uint8 class.

Matlab code *whos* used to determine the image class.

Table 9: *whos* source code

```
>> whos
  Name        Size              Bytes  Class     Attributes
  A           538x800x3         1291200        uint8
```

If the image is a binary image (black and white) the x-axis of the histogram only got a value of 2 as shown on figure 23.

For example:

Table 10: Image to black white code

```
>> a = imread('C:\Users\farisms\Pictures\Bachalpseeflowers.jpg');
>> c= im2bw(a);
>> subplot(2,1,1), imshow(c), title('Black and White image')
>> subplot(2,1,2), imhist(c), title('Histogram of Black and White image')
```

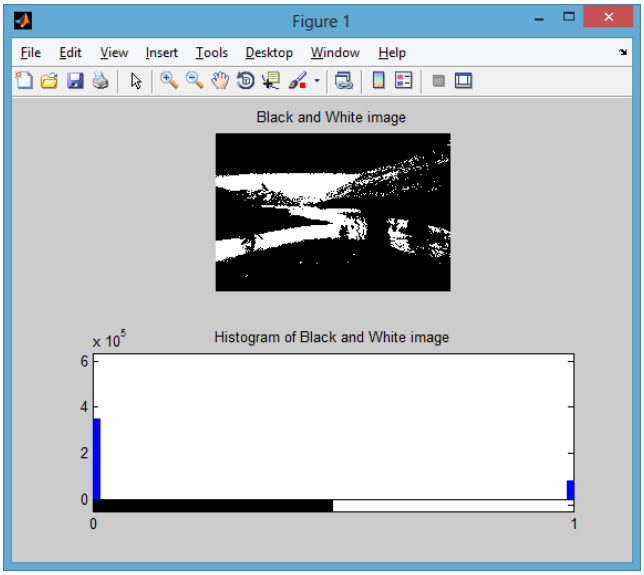


Figure 23: Black and white image with it histogram

4.1.3 Separate Red, Green and Blue component

The Matlab code on how to 3 channels of red, green and blue from colour image is shown belows. The output from the Matlab code can be seen on figure 24. Figure 25 shows the histogram represent the each color channel.

Table 11: Separate Red, Green and Blue component code

```
a = imread('C:\Users\FARISMS\Pictures\Red_Lory_(Eos_bornea)-6.jpg');
aRed = a(:,:,1);
aGreen = a(:,:,2);
aBlue = a(:,:,3);
subplot(2,2,1), imshow(a); title('Ori photo');
subplot(2,2,2), imshow(aRed); title('Red');
subplot(2,2,3), imshow(aGreen); title('Green');
subplot(2,2,4), imshow(aBlue); title('Blue');
```

Figure 24: Separated red, green, and blue channels

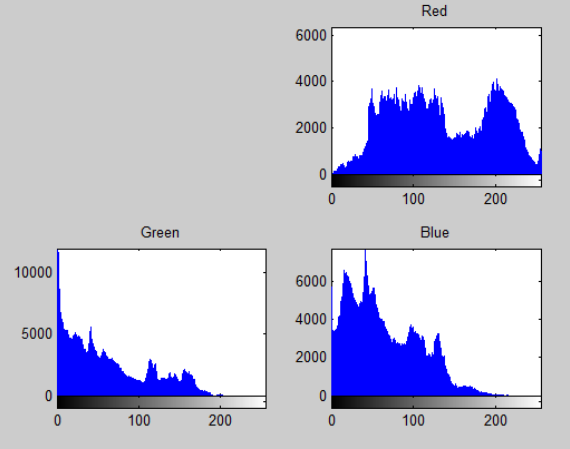


Figure 25: Histogram for red, green, and blue channels

4.2 Result on ulcer assessment

4.2.1 Color Correction

Figure 26 shows an image of ulcer before and after the color correction. Color correction need to be done in order to synchronise the entire ulcer image. This synchronisation is based on the histogram of reference white sticker which was attached initially on the ulcer surface. The color correction is needed for every ulcer image that was taken before started the analysis the ulcer image to provide more accurate result.

(b) Before color correction (a) After color correction

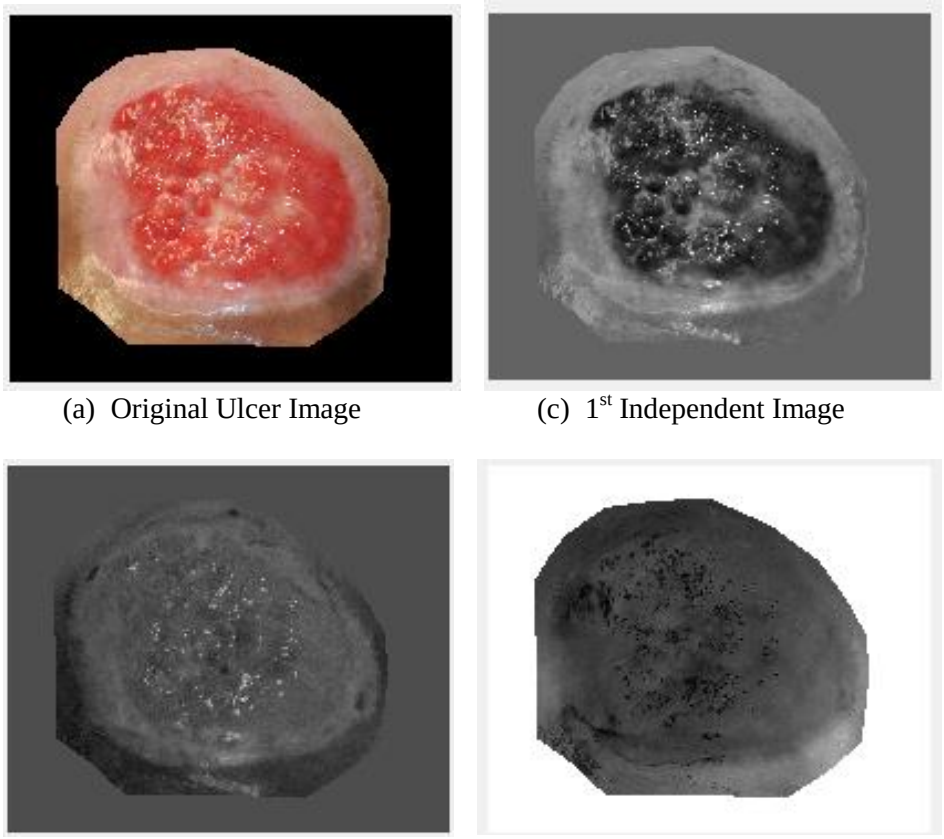
Figure 26: Ulcer image before and after color correction

4.2.2 Granulation tissue detection

Figure 27 shows the several independent components that have been extracted from color images that have been simulated using Matlab. From the original image (after color correction), the ICA has been implement and give the random output of several independent components images that appear on the ulcer surface such as melanin, necrosis tissue and granulation tissue.

Figure 27(c) shows the first independent source image which represents areas of significant intensity value range (dark region on the image) that can be clearly distinguished from the rest of the image. These areas indicate haemoglobin distribution which reflects regions of granulation tissue. The other independent image is a random

independent component that has been extracted by ICA. If the independent image that represents haemoglobin distribution is not defined on first run, then second or third run is required until the haemoglobin distribution image was defined.



(a) Original Ulcer Image

(c) 1st Independent Image

(b) 2nd Independent Image

(d) 3rd Independent Image

Figure 27 : Extracted independent sources from observed colour ulcer image

4.2.3 Segmentation of granulation tissue

Figure 28 shows the result obtains from classifying an extracted grey level haemoglobin image into clusters. Then, the classified image was converted into a binary image based on the intensity values of the clustered granulation tissue region. Figure 29 shows the result of converting the classified image in figure 28 into a binary image with segmented regions of granulation tissue.

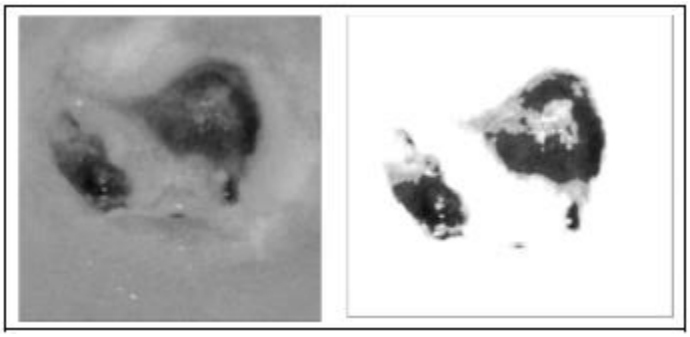


Figure 28: Classified Image obtained from haemoglobin image

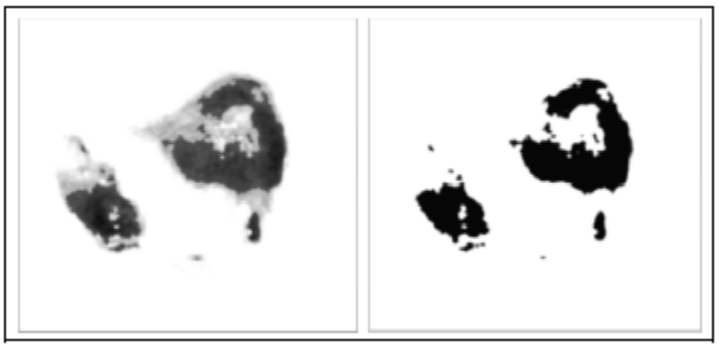


Figure 29: Binary image obtained from classified image

4.3 Android Application

4.3.1 Working Mechanism

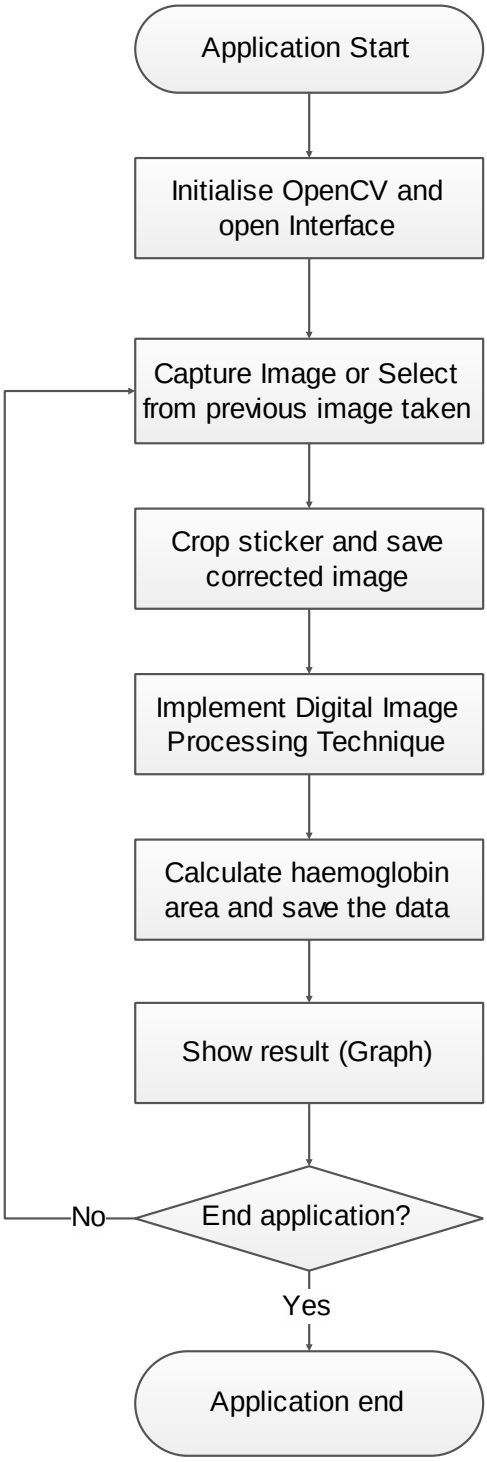


Figure 30: Basic project android application working mechanism

Figure 30 shows the basic project android application working mechanism. After the application start, the OpenCV will be initialise. This is to make sure the OpenCV manager is pre-installed in android smartphone. The initialisation will download and install OpenCV manager if there is no OpenCV manager in the phone. Source code below on table 7 shows how to initialise the OpenCV manager.

The latest version of the OpenCV manager on android is about 30MB (MegaBytes) in sizes and it's not too big. Almost all android smartphone can download and install OpenCV manager because the android smartphone will come with at least 256MB of internal storage.

Table 12 : Source code for BaseLoaderCallback to initialise OpenCV

```
public class MyActivity extends Activity implements HelperCallbackInterface
{
private BaseLoaderCallback mOpenCVCallBack = new BaseLoaderCallback(this) {
  @Override
  public void onManagerConnected(int status) {
    switch (status) {
      case LoaderCallbackInterface.SUCCESS:
      {
        Log.i(TAG, "OpenCV loaded successfully");
        // Create and set View
        mView = new puzzle15View(mAppContext);
        setContentView(mView);
      } break;
      default:
      {
        super.onManagerConnected(status);
      } break;
    }
  }
};

/** Call on every application resume **/
@Override
protected void onResume()
{
   Log.i(TAG, "Called onResume");
   super.onResume();

   Log.i(TAG, "Trying to load OpenCV library");
   if (!OpenCVLoader.initAsync(OpenCVLoader.OPENCV_VERSION_2_4_6, this,
mOpenCVCallBack))
   {
      Log.e(TAG, "Cannot connect to OpenCV Manager");
   }
}
```

After initialise the OpenCV, the interface will shows up.

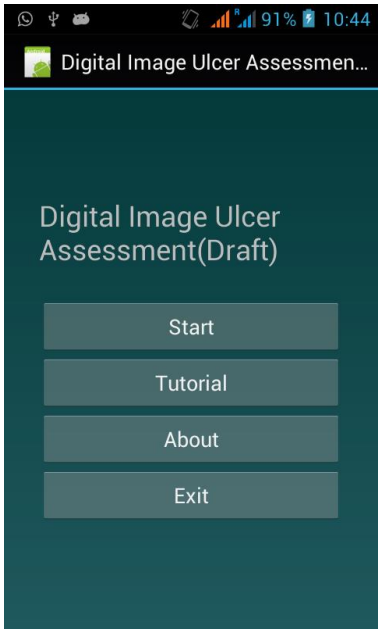


Figure 31: Digital Image Ulcer Assessment interface

The figure 31 shows the graphic user-interface (GUI) of Digital Image Ulcer Assessment on Android application. This GUI will come up first when opening the application. The interface source file is inside the res/layout/main.xml of a project directory.

Table 13: Source code inside main.xml layout

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:background="@color/background"
  android:gravity="center"
  android:padding="30dip">
  <TextView
    android:text="@string/main_title"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:layout_marginBottom="25dip"
    android:textSize="24.5sp" />
  <Button
```

```
    android:id="@+id/start_button"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/start_label" />
  <Button
    android:id="@+id/tutorial_button"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/tutorial_label" />
  <Button
    android:id="@+id/about_button"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/about_label" />
  <Button
    android:id="@+id/exit_button"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/exit_label" />
</LinearLayout>
```

The function *text view* on the java code above was used to view the text "Digital Image Ulcer Assessment (Draft)" on the GUI. The function <*button*> used to insert a button on the GUI. Every buttons was represented by an *id* each, which provided an identity to each button.

4.3.2 Button Interaction

These 4 buttons react according to the specific function.

1) Start button

Start button would navigate us into 2 options either "capture a photo" or "choose image from SD card". First option told us to capture the ulcer image of the patient, while the second option told use to select the already present ulcer image on the SD card.

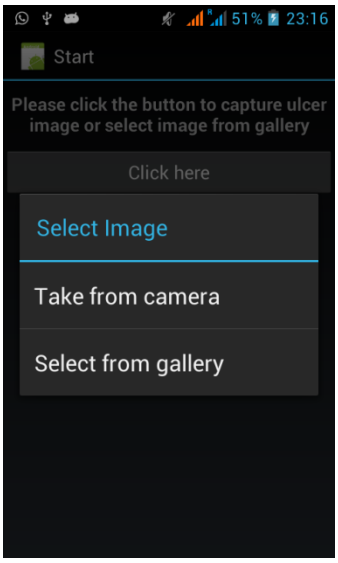


Figure 32: Intent dialog, choose either capture a picture or select from gallery

Code below shows an intent call a camera to capture a photo.

```
Intent captureIntent      = new Intent(MediaStore. ACTION_IMAGE_CAPTURE);
```

The camera feature need to be declare on the androidmanisfest.xml file to give a access to the access the camera

```
<uses-feature android:name="android.hardware.camera" />
```

A code shows below used to get image from gallery

Action function *ACTION_GET_CONTENT* used to open the folder directory.

```
Intent intent = new Intent();
intent.setType("image/*");
intent.setAction(Intent.ACTION_GET_CONTENT);
```

The application need to have a permission to access the SD card, so the code below was applied on the AndroidManifest.xml file.

```
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Then the ulcer image was cropped using this function code as shown in Table 14. White sticker on ulcer image was cropped in order to use as a reference to correct the ulcer image; color correction.

Table 14: Crop image function code

```
private void doCrop() {
    final ArrayList<CropOption> cropOptions = new ArrayList<CropOption>();

    Intent intent = new Intent("com.android.camera.action.CROP");
    intent.setType("image/*");

    List<ResolveInfo> list = getPackageManager().queryIntentActivities( intent, 0 );

    int size = list.size();

    if (size == 0) {
        Toast.makeText(this, "Cannot find image crop app", Toast.LENGTH_SHORT).show();
        return;
    } else {
        intent.setData(mImageCaptureUri);
        intent.putExtra("outputX", 200);
        intent.putExtra("outputY", 200);
        intent.putExtra("aspectX", 1);
        intent.putExtra("aspectY", 1);
        intent.putExtra("scale", true);
        intent.putExtra("return-data", true);

        if (size == 1) {
            Intent i        = new Intent(intent);
            ResolveInfo res    = list.get(0);
            i.setComponent( new ComponentName(res.activityInfo.packageName,
res.activityInfo.name));
            startActivityForResult(i, CROP_FROM_CAMERA);
        } else {
            for (ResolveInfo res : list) {
                final CropOption co = new CropOption();
                co.title  =
getPackageManager().getApplicationLabel(res.activityInfo.applicationInfo);
                co.icon        =
getPackageManager().getApplicationIcon(res.activityInfo.applicationInfo);
                co.appIntent= new Intent(intent);
                co.appIntent.setComponent( new
ComponentName(res.activityInfo.packageName, res.activityInfo.name));
                cropOptions.add(co);
            }

            CropOptionAdapter adapter = new CropOptionAdapter(getApplicationContext(),
cropOptions);
```

```
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setTitle("Choose Crop App");
            builder.setAdapter( adapter, new DialogInterface.OnClickListener() {
                public void onClick( DialogInterface dialog, int item ) {
                        startActivityForResult( cropOptions.get(item).appIntent,
CROP_FROM_CAMERA);
                }
            });

            builder.setOnCancelListener( new DialogInterface.OnCancelListener() {
                @Override
                public void onCancel( DialogInterface dialog ) {
                        if (mImageCaptureUri != null ) {
                                getContentResolver().delete(mImageCaptureUri, null, null );
                                mImageCaptureUri = null;
                        }
                }
            } );

            AlertDialog alert = builder.create();
            alert.show();
        }
    }
}
```

2) Tutorial button

Basically, *tutorial* button is just for the tutorial to this application. This button will navigate user to open a webpage, which is a tutorial webpage. In the tutorial webpage, there will be a video and step-by-step on how to operate the application.

Example code below shows how the application can interact with the webpage *www.google.com*.

```
Uri uri = Uri.parse("http://www.google.com");
Intent t = new Intent(Intent.ACTION_VIEW, uri);
startActivity(t);
```

3) About button

An *about* button will tell the user all about the digital image ulcer assessment application on android, the developer and also an email of developer if there any enquiries to ask. Figure 33 shows the about dialog.

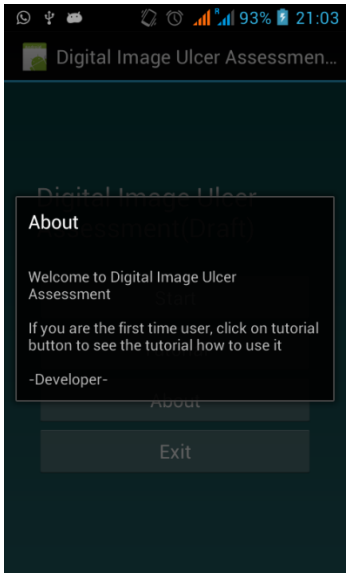


Figure 33: About dialog

4) Exit button

Exit button as it name, it will exit the application when the user press the button. Before closing the button, it will automatically save all the important data.

4.3.3 Implement of Digital Image Processing

Extraction of RGB Channel

Basically, the image of android will be decoded on YUV color if using basic read image method. The source code below on table 9 shows how to convert the YUV into RGB code color.

Table 15: Source code to convert YUV into RGB code color

```
static public void decodeYUV420SP(int[] rgb, byte[] yuv420sp, int width, int height) {
        final int frameSize = width * height;

        for (int j = 0, yp = 0; j < height; j++) {
            int uvp = frameSize + (j >> 1) * width, u = 0, v = 0;
            for (int i = 0; i < width; i++, yp++) {
                int y = (0xff & ((int) yuv420sp[yp])) - 16;
                if (y < 0) y = 0;
                if ((i & 1) == 0) {
                    v = (0xff & yuv420sp[uvp++]) - 128;
                    u = (0xff & yuv420sp[uvp++]) - 128;
                }

                int y1192 = 1192 * y;
                int r = (y1192 + 1634 * v);
                int g = (y1192 - 833 * v - 400 * u);
                int b = (y1192 + 2066 * u);

                if (r < 0) r = 0; else if (r > 262143) r = 262143;
                if (g < 0) g = 0; else if (g > 262143) g = 262143;
                if (b < 0) b = 0; else if (b > 262143) b = 262143;

                rgb[yp] = 0xff000000 | ((r << 6) & 0xff0000) | ((g >> 2) &
0xff00) | ((b >> 10) & 0xff);
            }
        }
    }
```

FastICA

FastICA algorithms method was taken from UMF environment. The application will call the FastICA functions through this environment using JNI. The JNI was needed to call or link the UMF environment with the android application because the UMF source code is made of native language which is C/C++.

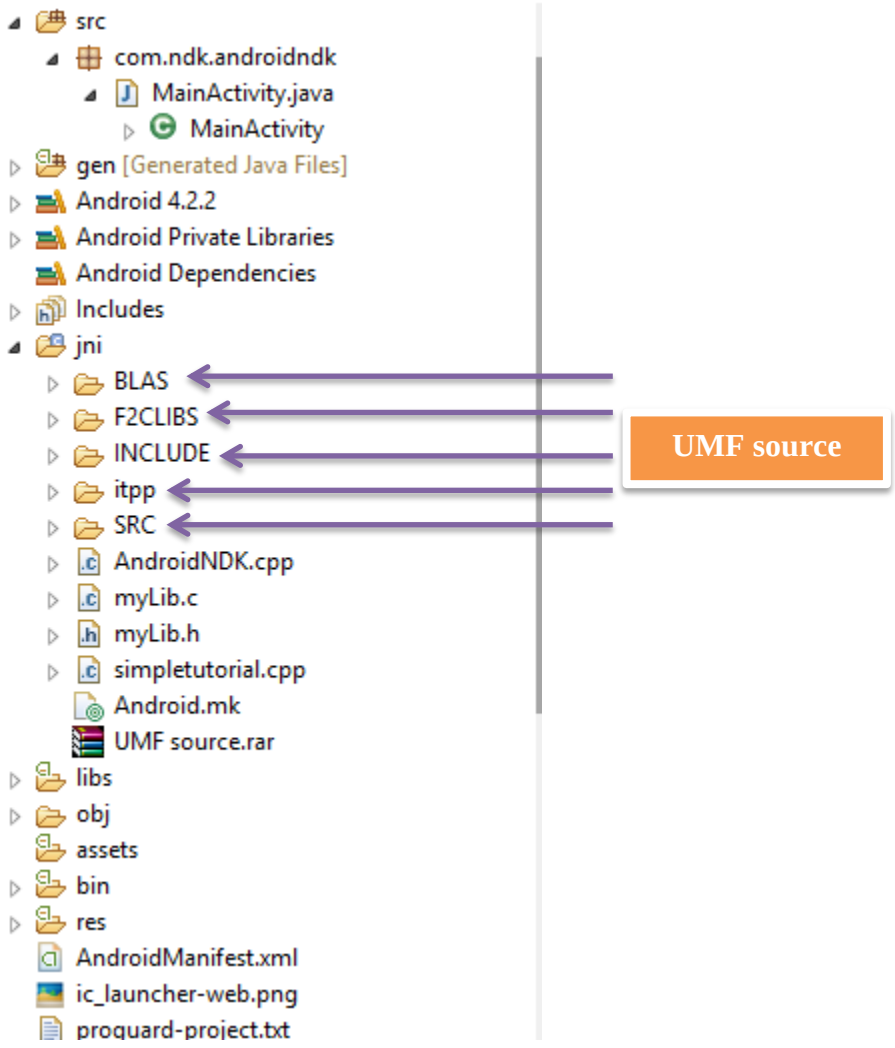


Figure 34: UMF sources files located inside android application

After the UMF sources files have been add to JNI folder, Android makefile was edited to merge the UMF environment with android application. Example of merging Android makefile can be seen on Appendix C.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

Chronic ulcers may result in severe paint to the patient. The doctors need to provide a proper treatment to treat the ulcer's patients. So the monitor of healing progression of ulcer is requirement in order to give a proper treatment. A Digital Image Processing technique is a technique use to make an assessment on the ulcers. A red granulation tissue on the ulcer image which contains haemoglobin pigment is a healing indicator. Digital image processing technique use an ICA to extract the red granulation tissue from the ulcer images because ICA has an ability to extract the heamoglobin (red granulation tissue) from the ulcer images. The digital image processing technique is a good technique for the ulcer assessment because the result provide is more precise, objective and reliable data.

Conventional tools use to capture an image for the digital image processing technique is a DSRL camera which is lack on portability and time consuming. An android smartphone was chosen as an alternative way to capture instead of DSLR camera, in the meantime analysis the image. By implementing the digital image processing technique inside a smartphone, it will be more efficient way to carry out the assessment.

Somehow, the implementation of the digital image processing technique on android application couldn't finish due to the time contrain. Based on the research project, author believes that the usage of digital image processing technique for ulcer assessment is feasible to use on android application.

The usage of Android NDK has an advantage to use OpenCV (Open Source Computer Vision) and UMF (UTP-Matlab like Framework) environment inside an android application. OpenCV provided a lot of computer vision library and it is small in size which enables it to be installing in all android smartphone. Meanwhile, with the usages of UMF environment, the FastICA algorithm would be ease to implement. Furthermore, the UMF environment provides a framework to convert some of Matlab code to C/C++. Both combinations are good practice almost for all digital images processing.

On the research project, the native language was used in order to create the android application with java language. High understanding in native language is an advantage

for those who want to continue this project. Furthermore, based on author experience, the implementation of Digital Image Processing Technique on android application would cost not less than RM15k. Author's believes that this research project will provide a road to a bright future for those who are interest in it.

CHAPTER 6: REFERENCES

[1] V. Ngan. Leg Ulcer. Available: http://dermnetnz.org/site-age-specific/leg-ulcers.html

[2] A. F. M. Hani, L. Arshad, A. S. Malik, A. Jamil, and F. Y. B. Bin, "Detection and classification of granulation tissue in chronic ulcers," in *Visual Informatics: Sustaining Research and Innovations*, ed: Springer, 2011, pp. 139-150.

[3] A. F. M. Hani, L. Arshad, A. S. Malik, A. Jamil, and F. Y. B. Bin, "Haemoglobin distribution in ulcers for healing assessment," in *Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on*, 2012, pp. 362-367.

[4] S. Loveridge. (2013, Third of world population will use a smartphone by 2017. Available: http://www.trustedreviews.com/news/third-of-world-population-will-use-a-smartphone-by-2017

[5] J. Sarasohn-Kahn, *How smartphones are changing health care for consumers and providers*: California HealthCare Foundation, 2010.

[6] C. A. Charles, A. F. Falabella, and A. C. Fernández-Obregón, "Chapter 31 - Leg ulcer management," in *Lower Extremity Soft Tissue & Cutaneous Plastic Surgery (Second Edition)*, G. D. Dockery and M. E. Crawford, Eds., ed Oxford: W.B. Saunders, 2012, pp. 447-469.

[7] N. Ouahes and T. J. Phillips, "Leg ulcers," *Current Problems in Dermatology,* vol. 7, pp. 114-142, 7// 1995.

[8] R. J. Goldman and R. Salcido, "More than one way to measure a wound: an overview of tools and techniques," *Advances in skin & wound care,* vol. 15, pp. 236-243, 2002.

[9] M. Herbin, A. Venot, J. Y. Devaux, and C. Piette, "Color quantitation through image processing in dermatology," *Medical Imaging, IEEE Transactions on,* vol. 9, pp. 262-269, 1990.

[10] A. Hyvärinen. (2013, What is Independent Component Analysis? Available: http://www.cs.helsinki.fi/u/ahyvarin/whatisica.shtml

[11] S. Vaseghi and H. Jetelová, "Principal and independent component analysis in image processing," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MOBICOM'06)*, 2006, pp. 1-5.

[12] A. Hyvarinen, P. Hoyer, and E. Oja, "Sparse code shrinkage: Denoising by nonlinear maximum likelihood estimation," *Advances in Neural Information Processing Systems,* pp. 473-479, 1999.

[13] Z. R. Godwin, J. C. Bockhold, L. Webster, S. Falwell, L. Bomze, and N. K. Tran, "Development of novel smart device based application for serial wound imaging and management," *Burns*.

[14] J. Morak, V. Schwetz, D. Hayn, F. Fruhwald, and G. Schreier, "Electronic data capture platform for clinical research based on mobile phones and Near Field Communication technology," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, 2008, pp. 5334-5337.

[15] J. Brenner. (2013, Pew Internet: Mobile. Available: http://pewinternet.org/Commentary/2012/February/Pew-Internet-Mobile.aspx

[16] K. Nagamine, "Apple Cedes Market Share in Smartphone Operating System Market as Android Surges and Windows Phone Gains, According to IDC " in *IDC Analyze the Future*, ed, 2013

[17] O. Postolache, P. S. Girao, M. Ribeiro, M. Guerra, J. Pincho, F. Santiago*, et al.*, "Enabling telecare assessment with pervasive sensing and Android OS smartphone," in *Medical Measurements and Applications Proceedings (MeMeA), 2011 IEEE International Workshop on*, 2011, pp. 288-293.

[18] C. Wimberly. Check your pulse with Instant Heart Rate. Available: http://androidandme.com/2010/08/applications/check-your-pulse-with-instant-heart-rate/

[19] Z. Tianyu, W. Lei, and M. Jun, "The exploitation and discussion of new mobile healthcare system model based on smart phone," in *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*, 2013, pp. 497-502.

[20] E. Burnette, *Hello, Android: Introducing Google's Mobile Development Platform ; [Android 2]*: "The" Pragmatic Bookshelf, 2010.

[21] R. Meier, *Professional Android 2 Application Development*: Wiley, 2010.

[22] (5 Dec 2013). Android NDK. Available: https://developer.android.com/tools/sdk/ndk/index.html

[23] F. Liu, *Android Native Development Kit Cookbook*: Packt Publishing, Limited, 2013.

[24] X. Zhu. (2011, Introduction to Android JNI development Using NDK - Part 1. Available: http://dotnetslackers.com/articles/net/Introduction-to-Android-JNI-development-Using-NDK-Part-1.aspx

[25] R. Rogers, J. Lombardo, Z. Mednieks, and G. B. Meike, *Android Application Development*: O'Reilly Media, 2009.

CHAPTER 7: APPENDICES

7.1 Appendix A : Main.java class

```
package org.example.draftproject;

import org.example.sudoku.R;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;

public class Main extends Activity implements OnClickListener {

  /** Called when the activity is first created. */
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);                    //always call superclass
    setContentView(R.layout.main);                                  //show main layout

    // Set up click listeners for all the buttons
    View startNewButton = findViewById(R.id.startnew_button);
    startNewButton.setOnClickListener(this);
    View startOldButton = findViewById(R.id.startold_button);
    startOldButton.setOnClickListener(this);
    View tutorialButton = findViewById(R.id.tutorial_button);
    tutorialButton.setOnClickListener(this);
    View aboutButton = findViewById(R.id.about_button);
    aboutButton.setOnClickListener(this);
    View exitButton = findViewById(R.id.exit_button);
    exitButton.setOnClickListener(this);
  }


/*************************************Play on/off music
  @Override
  protected void onResume() {
    super.onResume();
    Music.play(this, R.raw.main);
  }

  @Override
  protected void onPause() {
    super.onPause();
    Music.stop(this);
```

```
  }
*****************************************/

  public void onClick(View v) {
    switch (v.getId()) {
    case R.id.startnew_button:                                                    //Start new button
                    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                    startActivity(intent);
      break;
      // ...
    case R.id.startold_button:                                                    //Start old button
//          Imagecrop();

      Intent l = new Intent(this, LoadImage.class);
      startActivity(l);
//      final String [] items                 = new String [] {"Take from camera", "Select from gallery"};
        //choose take picture or select from gallery
//      ArrayAdapter<String> adapter          = new ArrayAdapter<String> (this,
android.R.layout.select_dialog_item,items);
//      AlertDialog.Builder builder           = new AlertDialog.Builder(this);
//
//      builder.setTitle("Select Image");
//      builder.setAdapter( adapter, new DialogInterface.OnClickListener() {
//
//      }



      break;
    // More buttons go here (if any) ...
    case R.id.tutorial_button:                                                    //tutorial button
      Intent t = new Intent(this, Tutorial.class);
      startActivity(t);
      break;
    case R.id.about_button:                                                       //About
button
      Intent a = new Intent(this, About.class);
      startActivity(a);
      break;
    case R.id.exit_button:                                                        //Exit button
          finish();
          break;

    }
  }

  /** option menu */
  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
    return true;
  }
  /*********************************************/
```

```
  @Override
  public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
    case R.id.settings:
      startActivity(new Intent(this, Prefs.class));
      return true;
    // More items go here (if any) ...
    }
    return false;
  }

}
```

7.2 Appendix B : Start.java class

```
package org.example.draftproject;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

import org.example.sudoku.R;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.ActivityNotFoundException;
import android.content.ComponentName;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.ResolveInfo;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

public class Start extends Activity {
        private Uri mImageCaptureUri;
        private ImageView mImageView;

        private static final int PICK_FROM_CAMERA = 1;
        private static final int CROP_FROM_CAMERA = 2;
        private static final int PICK_FROM_FILE = 3;

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.start_layout);

        final String [] items                   = new String [] {"Take from camera", "Select from gallery"};
            //choose take picture or select from gallery
                    ArrayAdapter<String> adapter        = new ArrayAdapter<String> (this,
android.R.layout.select_dialog_item,items);
                    AlertDialog.Builder builder                 = new AlertDialog.Builder(this);

                    builder.setTitle("Select Image");
                    builder.setAdapter( adapter, new DialogInterface.OnClickListener() {
                            public void onClick( DialogInterface dialog, int item ) { //pick from camera
                                    if (item == 0) {
                                            Intent captureIntent            = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);

                                            File imagesFolder = new
File(Environment.getExternalStorageDirectory(), "MyImages");
                                            imagesFolder.mkdirs(); // <----
                                            File image = new File(imagesFolder, "image_001.jpg");

                                            mImageCaptureUri = Uri.fromFile(image);


//                                          mImageCaptureUri = Uri.fromFile(new
File(Environment.getExternalStorageDirectory(),                 //save temp picture file
//                                                                      "tmp_avatar_" +
String.valueOf(System.currentTimeMillis()) + ".jpg"));


            captureIntent.putExtra(android.provider.MediaStore.EXTRA_OUTPUT, mImageCaptureUri);

                                            try {
                                                    captureIntent.putExtra("return-data", true);

                                                    startActivityForResult(captureIntent,
PICK_FROM_CAMERA);
                                            } catch (ActivityNotFoundException e) {
                                                    e.printStackTrace();
                                            }
                                    } else { //pick from file
                                            Intent getPicIntent = new Intent();

                      getPicIntent.setType("image/*");
                      getPicIntent.setAction(Intent.ACTION_GET_CONTENT);

                      startActivityForResult(Intent.createChooser(getPicIntent, "Complete action using"),
PICK_FROM_FILE);
                                    }
                            }
                    } );

                    final AlertDialog dialog = builder.create();

                    Button button      = (Button) findViewById(R.id.btn_crop);
                    mImageView                  = (ImageView) findViewById(R.id.iv_photo);
```

```
                button.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                                dialog.show();
                        }
                });
}

@Override
        protected void onActivityResult(int requestCode, int resultCode, Intent data) {
           if (resultCode != RESULT_OK) return;

           switch (requestCode) {
                        case PICK_FROM_CAMERA:
                                doCrop();

                                break;

                        case PICK_FROM_FILE:
                                mImageCaptureUri = data.getData();

                                doCrop();

                                break;

                        case CROP_FROM_CAMERA:
                           Bundle extras = data.getExtras();

                           if (extras != null) {
                              Bitmap photo = extras.getParcelable("data");

                              mImageView.setImageBitmap(photo);
                           }

                           File f = new File(mImageCaptureUri.getPath());

                           if (f.exists()) f.delete();

                           break;

           }
        }

private void doCrop() {
                final ArrayList<CropOption> cropOptions = new ArrayList<CropOption>();

        Intent intent = new Intent("com.android.camera.action.CROP");
   intent.setType("image/*");

   List<ResolveInfo> list = getPackageManager().queryIntentActivities( intent, 0 );

   int size = list.size();

   if (size == 0) {
        Toast.makeText(this, "Can not find image crop app", Toast.LENGTH_SHORT).show();
```

```
            return;
        } else {
            intent.setData(mImageCaptureUri);

            intent.putExtra("outputX", 200);
            intent.putExtra("outputY", 200);
            intent.putExtra("aspectX", 1);
            intent.putExtra("aspectY", 1);
            intent.putExtra("scale", true);
            intent.putExtra("return-data", true);

            if (size == 1) {
                    Intent i        = new Intent(intent);
                    ResolveInfo res    = list.get(0);

                    i.setComponent( new ComponentName(res.activityInfo.packageName,
res.activityInfo.name));

                    startActivityForResult(i, CROP_FROM_CAMERA);
            } else {
                        for (ResolveInfo res : list) {
                            final CropOption co = new CropOption();

                            co.title    =
getPackageManager().getApplicationLabel(res.activityInfo.applicationInfo);
                            co.icon        =
getPackageManager().getApplicationIcon(res.activityInfo.applicationInfo);
                            co.appIntent= new Intent(intent);

                            co.appIntent.setComponent( new
ComponentName(res.activityInfo.packageName, res.activityInfo.name));

                           cropOptions.add(co);
                        }

                        CropOptionAdapter adapter = new CropOptionAdapter(getApplicationContext(),
cropOptions);

                        AlertDialog.Builder builder = new AlertDialog.Builder(this);
                        builder.setTitle("Choose Crop App");
                        builder.setAdapter( adapter, new DialogInterface.OnClickListener() {
                            public void onClick( DialogInterface dialog, int item ) {
                                startActivityForResult( cropOptions.get(item).appIntent,
CROP_FROM_CAMERA);
                            }
                        });

                        builder.setOnCancelListener( new DialogInterface.OnCancelListener() {
                            @Override
                            public void onCancel( DialogInterface dialog ) {

                                if (mImageCaptureUri != null ) {
                                    getContentResolver().delete(mImageCaptureUri, null, null );
                                    mImageCaptureUri = null;
                                }
```

```
                        }
                    } );

                    AlertDialog alert = builder.create();

                    alert.show();
            }
        }
            }
}
```

7.3 Appendix C: Some part of Android.mk source code

```
#-------------------------------------------------------------------#

LOCAL_PATH := $(MY_TOP_PATH)
#top_srcdir := ./
include $(CLEAR_VARS)

include $(LOCAL_PATH)/itpp/sources.mk
include $(LOCAL_PATH)/itpp/base/sources.mk

include $(LOCAL_PATH)/itpp/signal/sources.mk


SOURCES_C := $(cpp_base_sources) $(cpp_signal_sources)

SOURCES_H := $(noinst_h_base_sources) $(h_base_sources) $(h_comm_sources) $(h_fixed_sources)  \
$(h_optim_sources) $(h_protocol_sources) $(h_signal_sources) $(h_srccode_sources) $(h_stat_sources)
#LOCAL_CFLAGS := -finline-functions#-Werror
LOCAL_CFLAGS := -march=armv7-a -mfpu=vfpv3 -mfloat-abi=softfp -Werror -g -Os
LOCAL_MODULE    := libitpp
LOCAL_C_INCLUDES := $(MY_TOP_PATH)/itpp/ \
                                        $(MY_TOP_PATH)/itpp/base \
                                        $(MY_TOP_PATH)/itpp/base/algebra \
                                        $(MY_TOP_PATH)/itpp/base/bessel \
                                        $(MY_TOP_PATH)/itpp/base/math \
                                        $(MY_TOP_PATH)/itpp/signal \
                                        $(MY_TOP_PATH)/itpp/stat \
                                        $(MY_TOP_PATH)/INCLUDE
#LOCAL_SRC_FILES := $(SOURCES_C)

LOCAL_SRC_FILES := itpp/base/bessel.cpp \
            itpp/base/binary.cpp \
            itpp/base/binfile.cpp \
            itpp/base/converters.cpp \
            itpp/base/copy_vector.cpp \
            itpp/base/fastmath.cpp \
            itpp/base/gf2mat.cpp \
```

```
itpp/base/help_functions.cpp \
itpp/base/itassert.cpp \
itpp/base/itcompat.cpp \
itpp/base/itfile.cpp \
itpp/base/mat.cpp \
itpp/base/matfunc.cpp \
itpp/base/operators.cpp \
itpp/base/parser.cpp \
itpp/base/random.cpp \
itpp/base/smat.cpp \
itpp/base/specmat.cpp \
itpp/base/svec.cpp \
itpp/base/timing.cpp \
itpp/base/vec.cpp \
itpp/base/algebra/cholesky.cpp \
itpp/base/algebra/det.cpp \
itpp/base/algebra/eigen.cpp \
itpp/base/algebra/inv.cpp \
itpp/base/algebra/ls_solve.cpp \
itpp/base/algebra/lu.cpp \
itpp/base/algebra/qr.cpp \
itpp/base/algebra/schur.cpp \
itpp/base/algebra/svd.cpp \
itpp/base/bessel/airy.cpp \
itpp/base/bessel/chbevl.cpp \
itpp/base/bessel/gamma.cpp \
itpp/base/bessel/hyperg.cpp \
itpp/base/bessel/i0.cpp \
itpp/base/bessel/i1.cpp \
itpp/base/bessel/iv.cpp \
itpp/base/bessel/jv.cpp \
itpp/base/bessel/k0.cpp \
itpp/base/bessel/k1.cpp \
itpp/base/bessel/kn.cpp \
itpp/base/bessel/polevl.cpp \
itpp/base/bessel/struve.cpp \
itpp/base/math/elem_math.cpp \
itpp/base/math/error.cpp \
itpp/base/math/integration.cpp \
itpp/base/math/log_exp.cpp \
itpp/base/math/misc.cpp \
itpp/base/math/trig_hyp.cpp \
itpp/signal/fastica.cpp \
itpp/signal/filter_design.cpp \
itpp/signal/filter.cpp \
itpp/signal/freq_filt.cpp \
itpp/signal/poly.cpp \
itpp/signal/resampling.cpp \
itpp/signal/sigfun.cpp \
itpp/signal/source.cpp \
itpp/signal/transforms.cpp \
itpp/signal/window.cpp     \
        itpp/stat/misc_stat.cpp \
itpp/stat/mog_diag.cpp \
itpp/stat/mog_diag_em.cpp \
itpp/stat/mog_diag_kmeans.cpp \
```

```
        itpp/stat/mog_generic.cpp
LOCAL_STATIC_LIBRARIES := liblapack libblas libf2c

LOCAL_CPPFLAGS += \
-std=gnu++0x -dM

LOCAL_LDLIBS := -llog
LOCAL_ARM_MODE := arm

include $(BUILD_STATIC_LIBRARY)
#-----------------------------------------------------------------------#
```