"MelScan: An Android Application for Early Detection of Melanoma"

By

Edwin Anak Nyambang 16738

A project dissertation submitted to the
Information and Communication Technology Program

In partial fulfillment of the requirement for the
BACHELOR (Hons) OF TECHNOLOGY
(INFORMATION & COMMUNICATION TECHNOLOGY)

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

MelScan: An Android Application for Early Detection of Melanoma

By

Edwin Anak Nyambang

A project dissertation submitted to the
Information & Communication Technology Program
Universiti Teknologi PETRONAS
In partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION & COMMUNICATION TECHNOLOGY)

Approved:

____
(DR.YONG SUET PENG @ VIVIAN)

Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

September 2014

CERTIFICATION OF ORIGINALITY

This was to certify that I am responsible for the work submitted in this project, that the original work was 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.

____

EDWIN ANAK NYAMBANG

ABSTRACT

Amongst the medical community, more focus was given on early detection of cancer, in particular melanoma, a type of skin cancer. Malignant melanoma was currently the leading source of death from skin diseases (Friedman et al., 2008).

This project was aimed at making a mobile application that can be used by civilians in the rural areas to detect melanoma at its early stages, because it was difficult for them to get help from dermatologists in rural areas, as well as getting access to the proper equipment for melanoma detection.

Due to time constraints, the scope of this study was narrowed to analyzing current image processing technology used in melanoma detection, especially on the Android mobile platform, developing an Android application that can use one of the image processing algorithms for melanoma detection, and to improve accuracy of melanoma detection, in particular during its early stages.

To do this, a market research was carried to find out what were the available applications that were out there in the Google Playstore (also known as Google Market) that was also used for early detection of melanoma, before comparing the types of available image processing algorithms that was to be used during the development phase.

Based on the findings, there were only 2 applications in the Google Market that was used for melanoma detection, and that the Prewitt operator makes for a good compromise between performance and complexity in developing the application.

Overall, the application was developed successfully, and in the future, more functions would be added to make this application more marketable, such as cloud storage function and wearable technology

CHAPTER 1

INTRODUCTION

1.1 Background of study

Melanoma iss a type of skin cancer, and it contributes to approximately 1.6% of the total number of cancer cases globally (Arroyo et al, 2013). As such, early detection and removal was the decisive factor to ensure high survival rates among patients. This was evidenced by Bono (1999), who noted that due to the fact that this disease can be diagnosed very early in its development, many physicians and specialists were trained to determine melanoma symptoms earlier and more effectively.

1.2 Problem statement

Malignant melanoma was the leading cause of death from skin diseases (Friedman et al., 2008). As also explained by Friedman, there were two standard methods for reducing morbidity and mortality from this skin disease. The first was identifying and removing the tumors directly, while the second was by early detection and removal of the tumor at an early stage. Normally, the second method was preferred because early removal guarantees a high chance of survival.

According to Friedman, patients can also participate in the identification of early malignant melanomas by using a through, easily taught method of self-examination. However, in today's modern society, technology makes almost everything automated, and this also includes melanoma detection. Unfortunately, this technology was still in its early stages, particularly in terms of portable machines designed for detecting and diagnosing malignant melanoma.

1.3 Objective and scope of study

The main objective of this research was:

- To analyze current technology in melanoma detection, especially Android applications
- To develop an Android application that uses the Prewitt operator in melanoma detection

- To improve melanoma detection, especially during the early stages

This research focuses on creating an Android application that utilizes the simplest algorithm for melanoma detection, and yet produces the most accurate diagnosis of the disease. The reasoning was that with a simple enough algorithm, processing the target (skin lesion) takes less time, and that with increased accuracy, may provide a minute, yet profound improvement in melanoma detection technology.

CHAPTER 2

LITERATURE REVIEW

2.1. Literature Review

One of the earliest and commonly used guide for melanoma diagnosis was the ABCD criteria (Bono et al, 1999), which stands for Asymmetry, Border, Color and Dimension respectively. While not highly effective for initial diagnosis of melanoma, this guide continues to present practical value in current clinical settings and thus becomes a recommended guideline for the early diagnosis of the disease.

To further expand on this guide, each potential lesion was visually determined to be benign (harmless) or malignant (dangerous) based on the ABCD criteria:

A – Asymmetry: The lesion in question was not equal or matched if an imaginary line was drawn through it.

B – Border: The borders of a melanoma lesion tend to be uneven, and its edges may be scalloped or notched.

C – Color: If the lesion in question has multiple shades of brown, tan or black, or if it was red, blue or other color, it was sufficient warning of a possible melanoma.

D – Diameter: Melanoma usually was larger than 6mm, but may appear smaller when first detected.

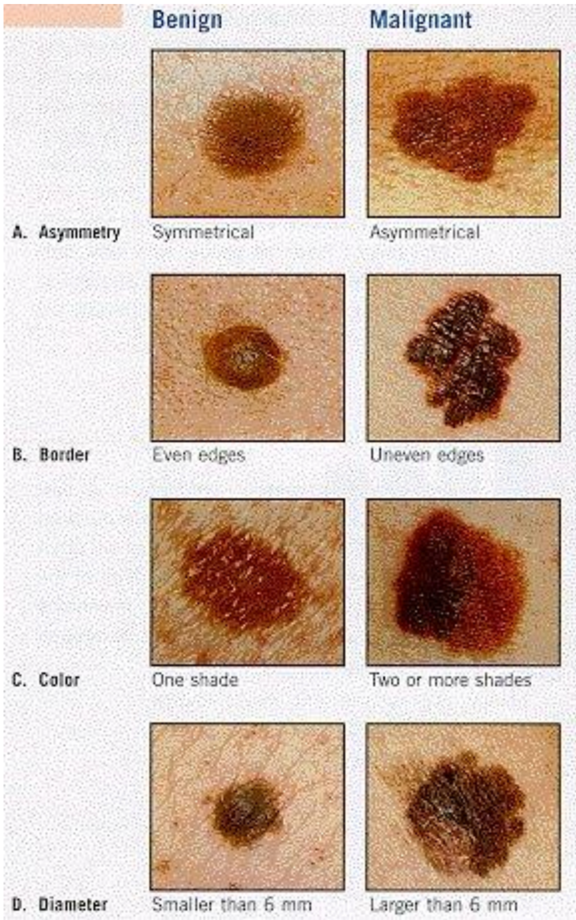


Fig.1 ABCD of melanoma

In image processing, one of the algorithms used to detect and calculate edges was the Prewitt operator. It was an edge detection algorithm, where it was simple enough for machine computation, but very sensitive to noise (Maini & Aggarwal, 2009).

Operator	Advantages	Disadvantages
Classical (Sobel, Prewitt, Kirsch,…)	• Simplicity • Detection of edges and their orientations	• Sensitivity to noise • Inaccurate
Zero Crossing (Laplacian, Second directional derivative)	• Detection of edges and their orientations • Having fixed characteristics in all directions	• Responding to some of the existing edges, • Sensitivity to noise
Laplacian of Gaussian(LoG) (Marr-Hildreth)	• Finding the correct places of edges, • Testing wider area around the pixel	• Malfunctioning at the corners, curves and where the gray level intensity function varies. • Not finding the orientation of edge because of using the Laplacian filter
Gaussian(Canny, Shen-Castan)	• Using probability for finding error rate, localization and response. • Improved signal to noise ratio • Better detection specially in noise conditions	• Complex computations, • False zero crossing, • Time consuming

Table 1. Comparison between existing image detection algorithms

In the global market, there were more Android smartphone users than iPhone users in the low and middle income group (Smith, 2013). This was due to the fact that it was cheaper, and yet can perform just as good, or better than an iPhone. This translates to easier access to the melanoma detection software created by this project, as more people can obtain it via the Android market.

This was further supported by the fact that the most popular smartphone operating system was Android, and that it runs on the widest selection of handheld devices at very affordable prices which goes down in value as newer models hit the market (Goldman, 2011). Moreover, because of time, budget and technical constraint, the system will not be implemented in real time and a prototype will be developed to act as a proof of concept and development

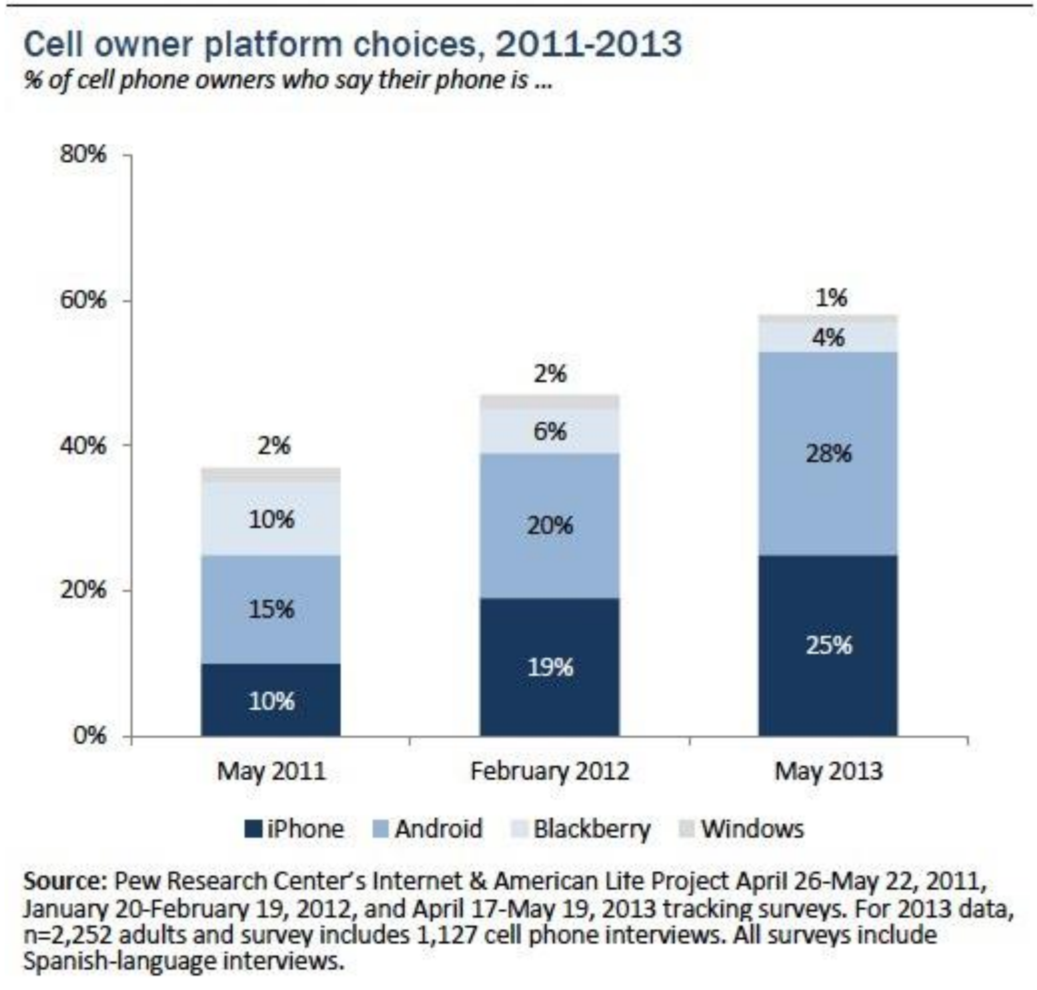


Table 2. Global possession of smartphones

In computer programming, Eclipse was an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages like C, C++, PHP and PERL. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

CHAPTER 3

METHODOLOGY

3.1. Methodology

This project was carried out using a qualitative method, where most of the project was carried based on reviews from other similar projects and research as mine and the project was carried to solve a problem from a point of view, namely to assist civilians in the rural areas to make early detection of melanoma

To begin with, a market research was conducted on whatever product was currently available on the market. To achieve this, a few criterias that need to be fulfilled for that product to be part of my project:

1. The product must be related to early detection of melanoma
2. The product in question was available for either Android or iOS devices, or both
3. The product in question must be able to perform early recognition of melanoma, using either the ABCDEs, the Ugly Duckling sign, or both
4. The product in question should be readily available, as in easily accessed for civilian use, with no additional hardware required to operate.

Based on these criteria, there were several applications that can conduct early detection of Melanoma. The first was SpotMole. Created by Cristian Munteanu in 2011, this application uses image processing and pattern recognition algorithms as its basis to detect melanoma.

The second application was Doctor Mole, created by Mark Shippen in 2013. This application also uses image processing and pattern recognition as its basis to melanoma detection.

Throughout this project, the Prewitt operator is applied as the image processing algorithm. The operator was used for edge detection in an image. It detects two types of edges which were vertical edges and horizontal edges. The reason was that it was simple enough to use to detect the edges of the lesion in question for analysis.

As such, there were a few steps involved in this application:

1. The image was captured by the hardware camera and stored in its memory for processing.
2. The image was run under the Prewitt operator to estimate the lesion border
3. Area inside the border was calculated to determine compactness of lesion
4. Algorithm was also used to calculate composition of color in lesion image
5. Results obtained from border estimation were compared with actual value obtained from reliable sources for determining whether lesion should be brought for further examination

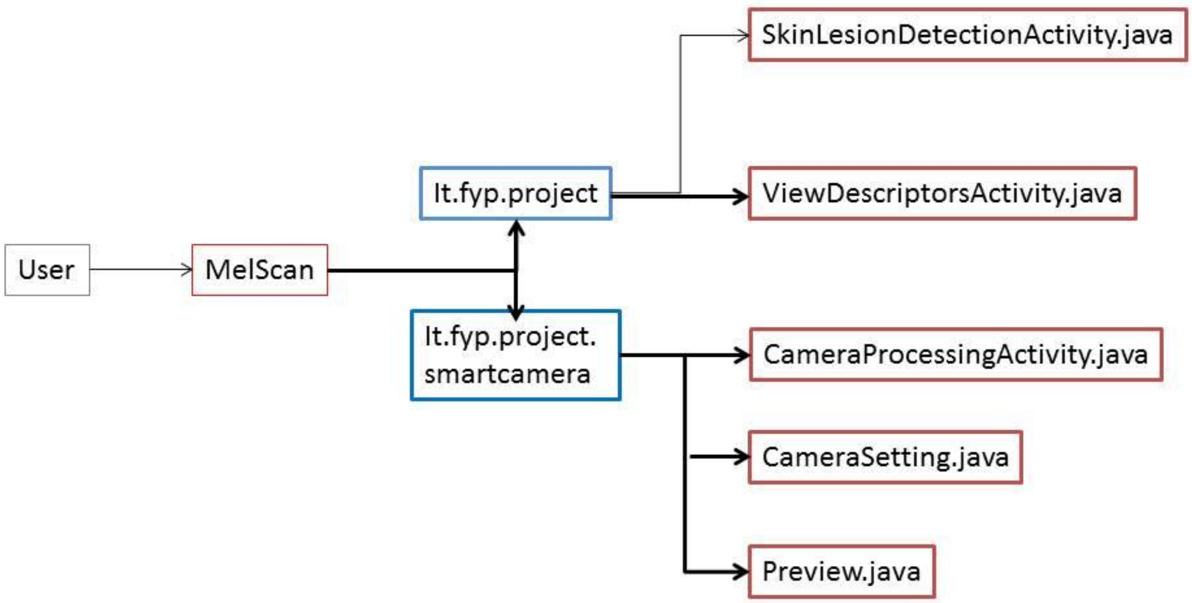


Fig.2 System design of MelScan

From the above figure, MelScan has been designed to be as simple as possible. For this purpose, the application has two separate packages, one will hold the main functions of MelScan (it.fyp.project), while the other holds the functions for the device where MelScan is to be housed (it.fyp.project.smartcamera).

In the it.fyp.project package, both the java class for the user interface (SkinLesionDetectionActivity.java) and the image processing algorithm (ViewDescriptorsActivity.java) are there, while the it.fyp.project.smartcamera package, it houses the java classes for camera processes (CameraProcessingActivity.java), camera settings (CameraSetting.java), and image preview (Preview.java).

3.1.1. Tools

In Eclipse IDE, the application uses the OpenCV function provided under the Imgproc package to run the Prewitt operator, as well as another algorithm to calculate the RGB color values of the image in question. Some of the notable methods under this function were described in the table below:

Method	Description
cvtColor (Mat src, Mat dst, int code, int dstCn)	Converts an image from one colour space to another.
Dilate (Mat src, Mat dst, Mat kernel)	Dilates an image by using a specific structuring element.
equalizeHist (Mat src, Mat dst)	Equalizes the histogram of a grayscale image.
filter2D (Mat src, Mat dst, int ddepth, Mat kernel, Point anchor, double delta)	Convolves an image with the kernel.
GaussianBlur (Mat src, Mat dst, Size ksize, double sigmaX)	Blurs an image using a Gaussian filter.
Integral (Mat src, Mat sum)	Calculates the integral of an image.

Table 3. Methods under OpenCV

During the development phase, Eclipse Juno, which is an integrated development environment (IDE) program, was used. The reason was that this program has various plugins to assist me in developing an Android application, as well as being light and easy to customize for me to integrate other Android development tools like software development kits (SDK) for older versions of Android as well as the current versions available in the market. This feature is very useful for creating an application that can operate in almost any version of Android.

In order to create an Android program for this project, it was insufficient to have Eclipse alone. This was because Eclipse IDE was designed to create Java applications. To solve this issue, Eclipse developers created various plugins for creating projects using different languages, such as C++ and PHP. For this project, the Eclipse IDE used for this project was equipped with the Android Development Tool (ADT) and software development kit (SDK) plugin, which allows the creation of a Java application that can run on Android smartphones.

Due to the rather complex nature of this application, multiple Java classes were created, which were then split into two different packages. The first package was “it.fyp.project”, which was used to hold the classes for the user interface. The second was “it.fyp.project.smartcamera”, which holds the classes for the camera interface of the application. Under the first class there were two Java files, which were SkinLesionDetectionActivity.java and ViewDescriptorsActivity.java. Both files were used when the user interacts with the user interface.

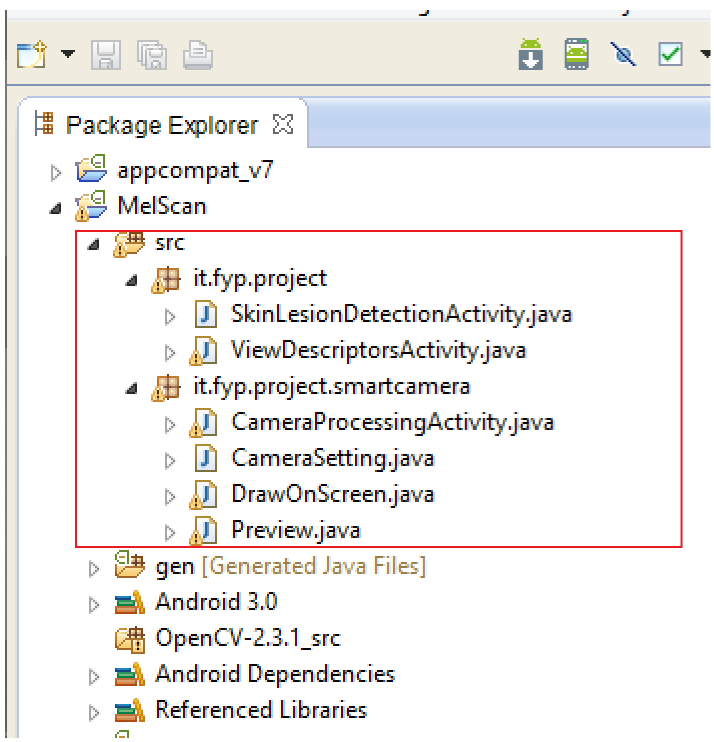


Fig 2. Package and classes used in project

When designing the user interface, Eclipse uses a drag-and-drop approach, where there were various widgets that users can choose and drop onto the interface. This will automatically generate its Extensible Markup Language (XML) so that users do not need to code everything, line by line.

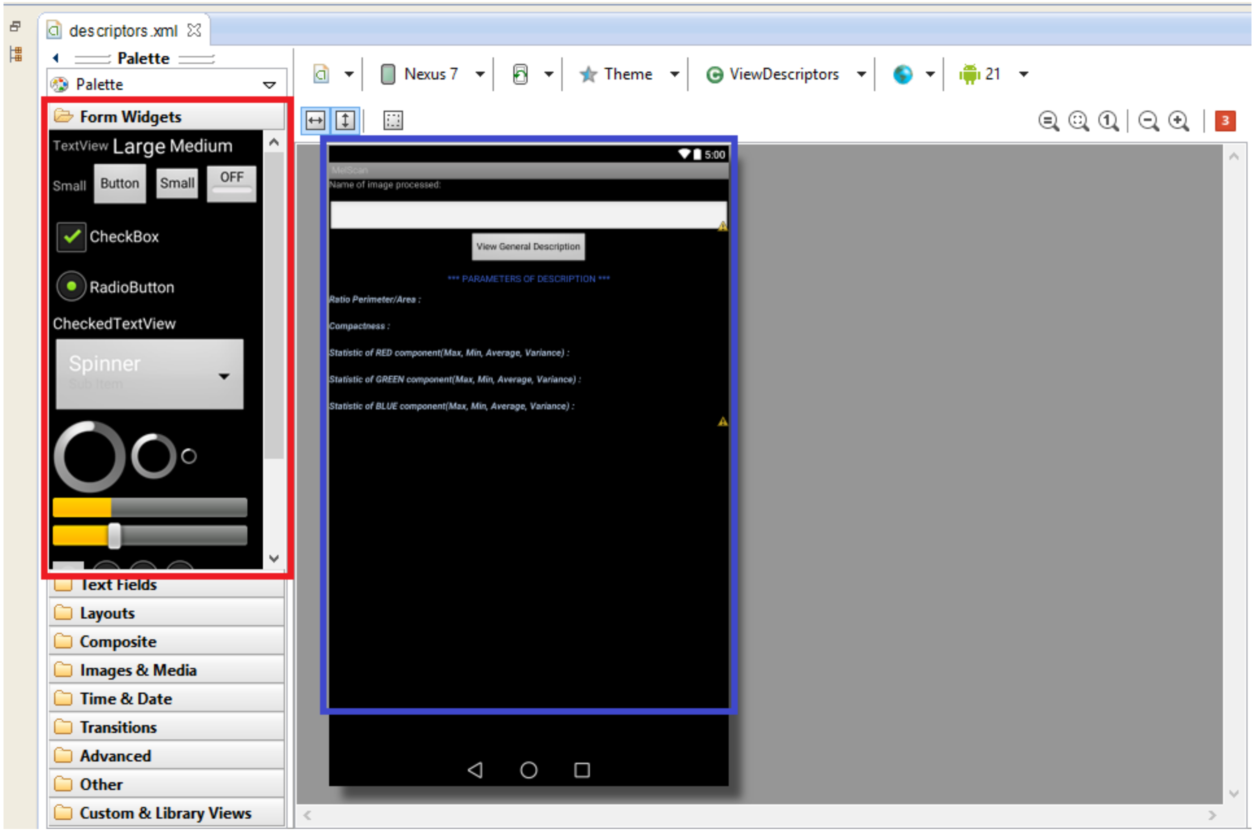


Fig 3. Drag-and-drop form widgets (red) and interface form (blue)

Fig 4. Eclipse-generated XML file

Once completed, the project was tested before put to use in the hardware. To accomplish this, Eclipse has an Android Virtual Machine plugin, where users can test run the project in a simulated Android smartphone on the computer.

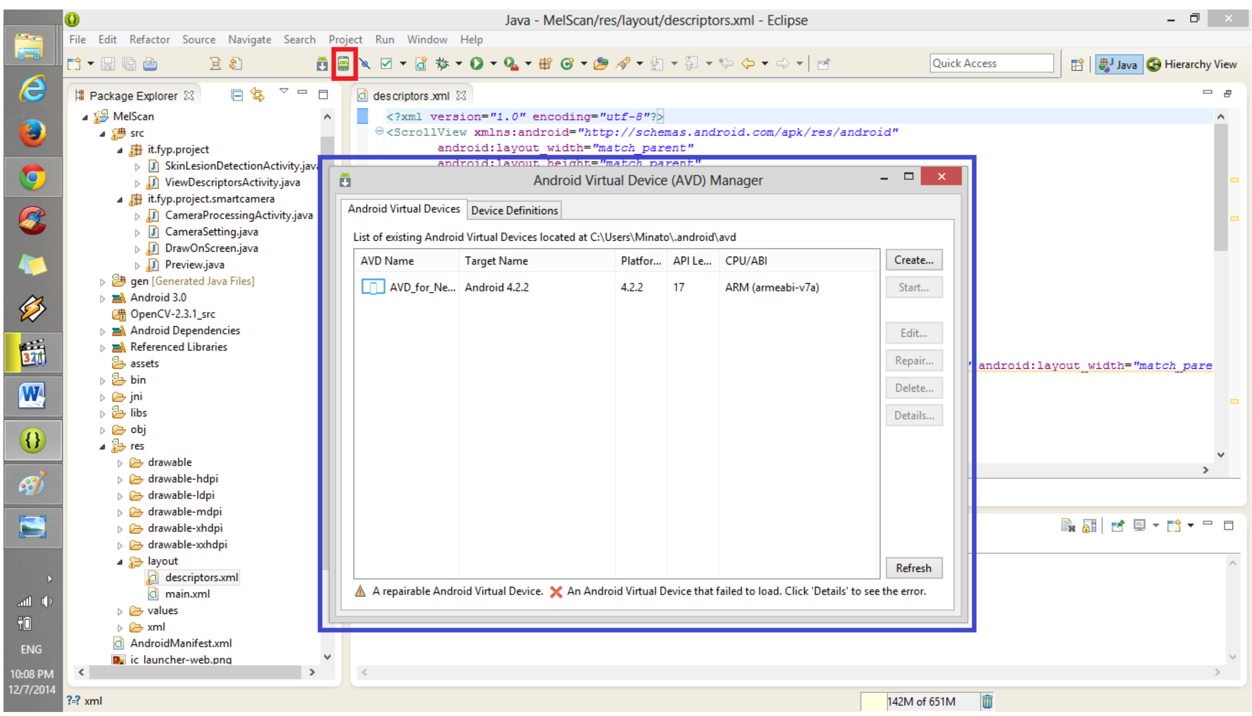


Fig 5. Android Virtual Machine icon (red) and interface (blue)

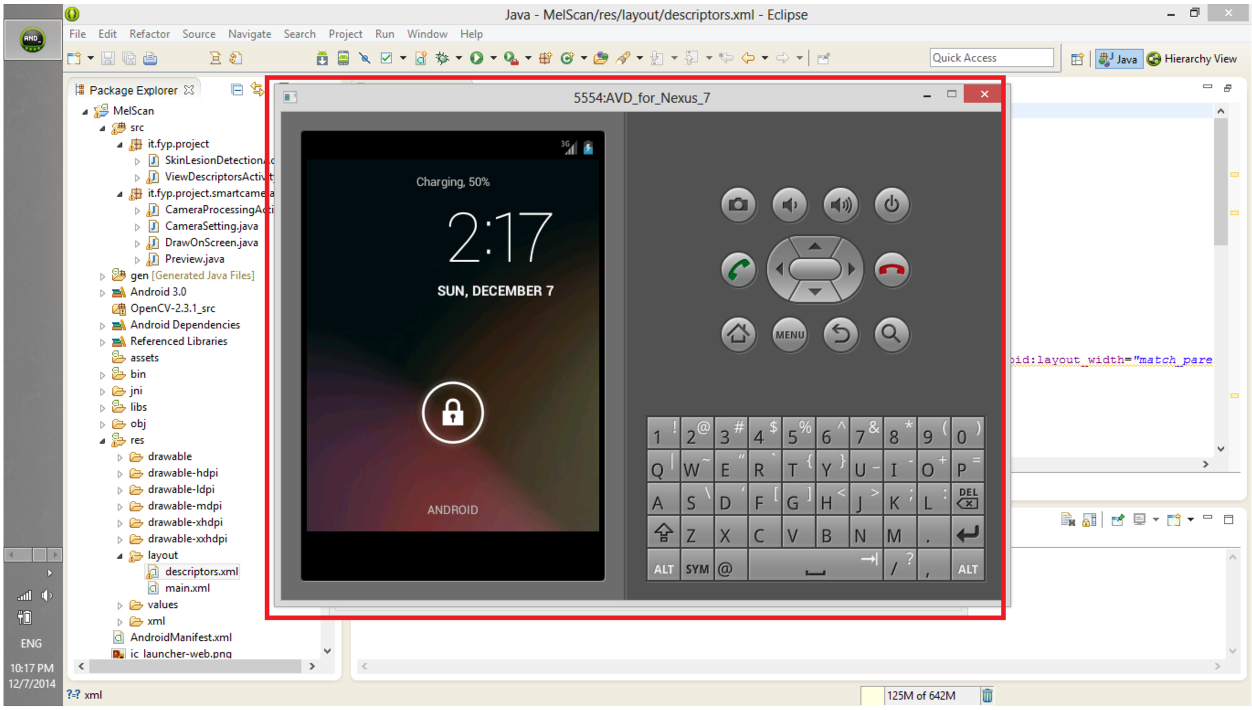


Fig 6. Virtual Device interface (red)

Once the interface was completed, the corresponding Java class is created, named "SkinLesionDetectionActivity.java". Included in this class were functions that enable the application to use the hardware camera, change camera settings and to analyze the image taken.

The main interface and code snippet was shown below:

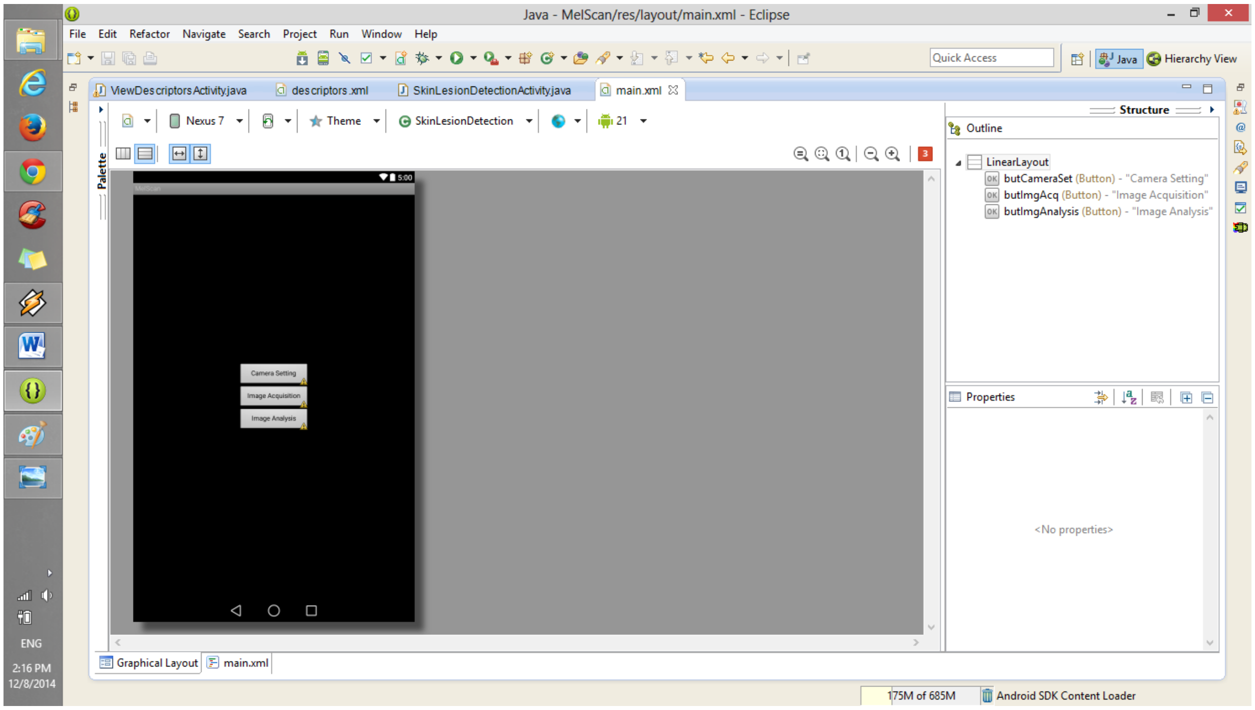


Fig 7. Main interface

ViewDescriptorsActivity.java | descriptors.xml | SkinLesionDetectionActivity.java | main.xml

```
Button butImgAcq = (Button) findViewById(R.id.butImgAcq);
butImgAcq.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        startCamera();
    }
});

Button butCameraSet = (Button) findViewById(R.id.butCameraSet);
butCameraSet.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        settingCamera();
    }
});

Button butImgAnalysis = (Button) findViewById(R.id.butImgAnalysis);
butImgAnalysis.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        viewLesionDescripion();
    }
});
```

Fig 8.Code snippet for button function in main interface

3.2 Gantt Chart and Key Milestones

	Week													
Details of work	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1. Initiation Phase														
1.1 Problem Identification														
1.2 Background of Study														
1.3 Project Approval														
1.4 Literature Review														
2. Research and Analysis Phase														
2.1 Market Research														
2.2 Tabulation of data														
2.3 Results and discussion														
2.4 Requirements gathering														
3. Design Phase														
3.1 Pseudocode generation														
3.2 Interface design														
3.3 System design														

= Key milestones

= Timeline

	Week													
Details of work	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**
1. Development Phase														
1.1 Interface design														
1.2 System prototyping														
1.3 Algorithm generation														
2. Testing Phase														
2.1 Alpha testing														
2.2 Beta testing														
2.3 Tabulation of results														
2.4 Discussion														

CHAPTER 4

RESULT

4.1. Result

Based on the market research carried out in the methodology, it can be seen that both these applications available in the market work differently from one another. Still, it was worth noting that these applications still have some common features.

As such, all results and comparison was tabulated below:

Product	Spotmole	Doctor Mole
Usage	• To create initial diagnosis of potential melanoma lesion	
Principle of operation	• Image processing	
Similarities	• Both use the ABCDE principle of melanoma detection as its diagnostic mechanism • Both applications were “freemiums”. In other words, basic functions were free, but users need to pay for advanced features	
Difference	• Spotmole has a high degree of false positives – almost anything dark on skin was considered as melanoma	• Doctor Mole does not give initial diagnosis, instead offers it as premium feature.
Pros	• Delivers quick results	• Fairly accurate evaluation of lesion

		based on the ABCD criteria
Cons	• Spotmole's output may be affected by images with poor contrast/brightness, color artifacts, uneven illumination, poor resolution, and skin hairs covering the mole	• When using application, viewfinder has very small zone to detect skin lesions.
User ratings	• 3.2 out of 5, from a total of 84 reviews	• 3.5 out of 5, from a total of 254 reviews

Table 4. Similarities between two current Android applications for melanoma detection

From this result, it can also be noted that not many people use this application, as the number of users for both applications were not many. This was because that this form of technology was still in its infancy. As such, it was still prone to give false results, which renders it sometimes unreliable to the users. However, the makers of these applications did stress out that their product was not meant to replace expert diagnosis, and thus this still point out that early judgment whether a lesion was considered a malignant melanoma lies on human decision, not machines.

Also, it was also worth noting that both applications were considered “freemiums”, where users can unlock more of its features by paying, usually via online transactions. This may prove to be a stumbling block, as interested individuals who want to use these applications may have to pay to use more of its special features, like online specialist support in the case of Spotmole, or full access to diagnostic results in the case of Doctor Mole.

The hardware required to run this application needs to have a few important functions, namely a camera to capture the image for analysis, as well as operating on Android. From the available budget allocated for students, an Asus Fonepad 7 tablet was purchased to house and run the application. The specification for the tablet was shown below:

Android version:	4.3 (Jelly Bean)
Display	7 inch LED Backlight WSVGA (1024x600)
Processor	2X2: Intel® Atom™ Multi-Core Z2520 Processor 1.2 GHz
Front Camera	0.3MP
Rear Camera	2.0 MP

Table 5. Hardware specification used in project

From the algorithm developed based on the Prewitt operator, the image in question was first captured by the tablet camera, where it was stored in its internal memory for image preprocessing. From there, the a copy of the image was created and converted from RGB into Grayscale, where the operator then compares the contrasting gradient of the image for approximation of the lesion border. After that, the operator proceeds to calculate the value of the perimeter and area, where the value was returned as a float value. As of now, there was no method to convert the value from float into round integers.

From the value returned in perimeter, the logic for evaluating the image was as follows:

If (diameter of lesion >= 0.635 cm)

tempTextV.setText (“The skin sample seems problematic. Please bring it to a dermatologist for further examination");

Else

tempTextV.setText (“There seems to be no problem found on the skin sample");

In Android programming, the “setText” function was the equivalent of the “printf” function in Java, which was to return a text value. The code snippet was shown below:

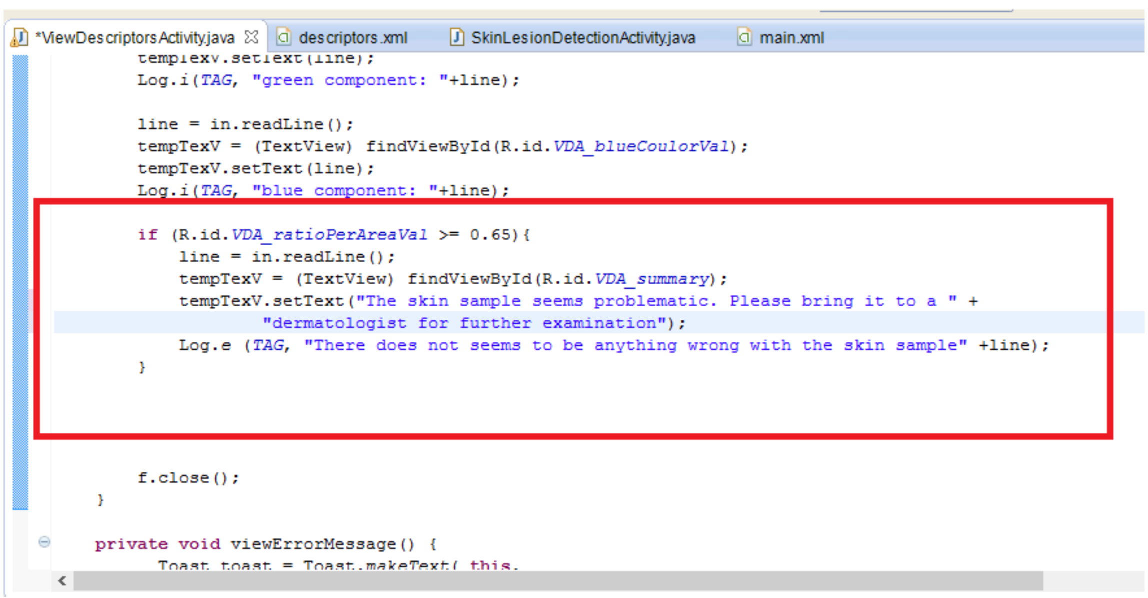


Fig 7. Code snippet for evaluation logic (red)

As a rule of thumb, when performing a self-diagnosis to detect melanoma, when the diameter of the lesion was larger than an eraser of a pencil (0.635cm), it was safe to assume that the lesion was a potential tumor and should be brought to an expert for further analysis. Also, if the border of the lesion was ragged, blurred or irregular, this also serves as a warning sign for potential melanoma.

Protoype design and testing

Fig 7. User interface

Fig 8. Sample image

Fig 9. Prewitt operator detecting and estimating perimeter of lesion (border)

Fig 10. Perimeter was transposed on lesion to show estimated size of lesion

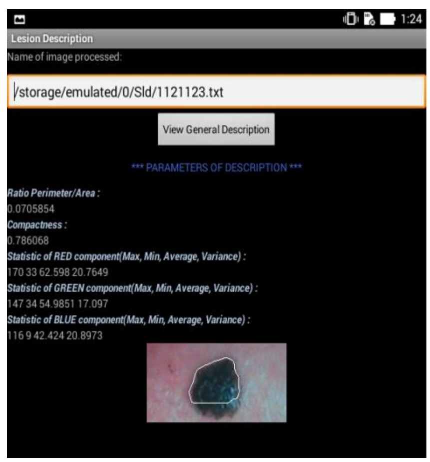


Fig 11. Algorithm gives numerical value of lesion for evaluation

During the testing phase, it has been noted that out of all 10 images used to test this application (8 malignant, 2 benign), all results show that the image samples were potential melanoma. This means that the edge detection algorithm was still not entirely accurate as it gives out a false positive when tested. The result of the test was attached in the appendix for reference.

CHAPTER 5

CONCLUSION

5.1. Conclusion

As a conclusion, MelScan has the potential to assist civilians in rural areas to perform early diagnosis of melanoma. In this matter, the Prewitt method was highly necessary to attain a balance between performance and computation time, as this method was efficient enough without causing a heavy load on the machine. This was highly important because the goal of this project was to produce an Android application that can assist in early detection of melanoma.

5.2. Future works and recommendation

Like all edge detection algorithms, the Prewitt operator was not without flaws. The major flaw in this operator was that it was very sensitive to external noise, such as vibrations generated from shaking hands. To propose that general practitioners bring a tripod whenever they go into rural areas for checkups was not feasible. As such, it was hoped that a better edge detection algorithm was used in place of the Prewitt operator without sacrificing hardware performance. Also, this application was expected to include a cloud storage function, where any images captured by the hardware camera was send to the online storage for analysis by experts worldwide.

CHAPTER 6

REFERENCES

1. Smartphone Ownership 2013. (n.d.). *Pew Research Centers Internet American Life Project RSS*. Retrieved July 20, 2014, from http://www.pewinternet.org/2013/06/05/smartphone-ownership-2013/

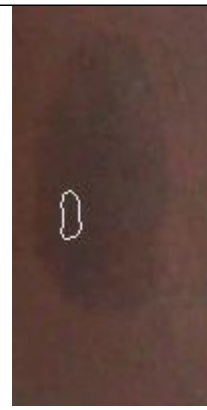
2. Bono, A., Tomatis, S., Bartoli, C., Tragni, G., Radaelli, G., Maurichi, A., et al. (2000, November 19). *The ABCD System of Melanoma Detection: A Spectrophotometric Analysis of the Asymmetry, Border, Color, and Dimension.* Retrieved August 3, 2014, from http://onlinelibrary.wiley.com/doi/10.1002/(SICI)10970142(19990101)85:1%3C72::AID-CNCR10%3E3.0.CO;2-Q/full#

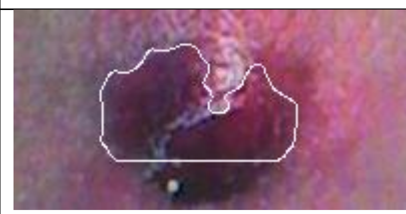
3. Friedman, R. J., Rigel, D. S., & Kopf, A. W. (2008, December 31). Early Detection of Malignant Melanoma; The Role of Physician Examination and Self-Examination of the Skin. *CA: A Cancer Journal for Clinicians. - Wiley Online Library.* Retrieved August 3, 2014, from http://onlinelibrary.wiley.com/journal/10.3322/(ISSN)1542-4863

4. Goldman, D. (2011, March 7). Android passes BlackBerry as No. 1 on smartphones.*CNNMoney*. Retrieved August 5, 2014, from http://money.cnn.com/2011/03/07/technology/android/

5. Maini, R., & Aggarwal, H. (2009). Study and Comparison of Various Image Edge Detection Techniques. *International Journal of Image Processing (IJIP), 3*(1).
6. J. L. G. Arroyo and B. G. Zapirain (2013, December 6). Detection of pigment network in dermoscopy images using supervised machine learning and structural analysis, Comput. Biol. Med, vol. 44. Retrieved August 5,2014 from http://dx.doi.org/10.1016/j.compbiomed.2013.11.002

APPENDIX

Sample Image	Lesion type	Result
	Benign	
	Benign	
	Malignant	
	Malignant	

	Malignant	
	Malignant	
	Malignant	
	Malignant	

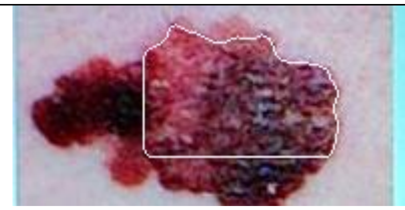
	Malignant	
	Malignant	

Fig 12.Image testing results for edge detection.

Fig 13a. Lesion analysis and result

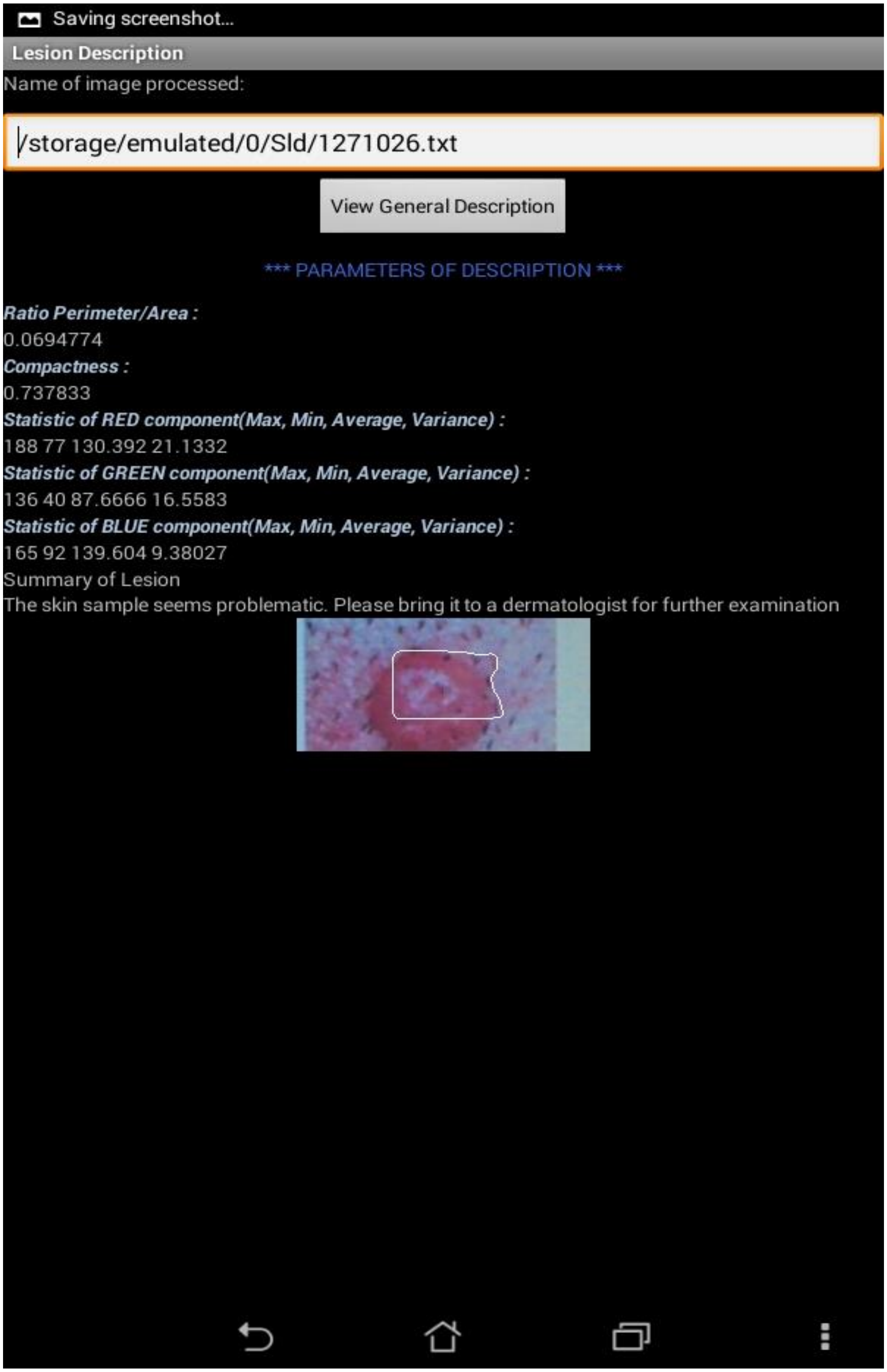


Fig 13b. Lesion analysis and result

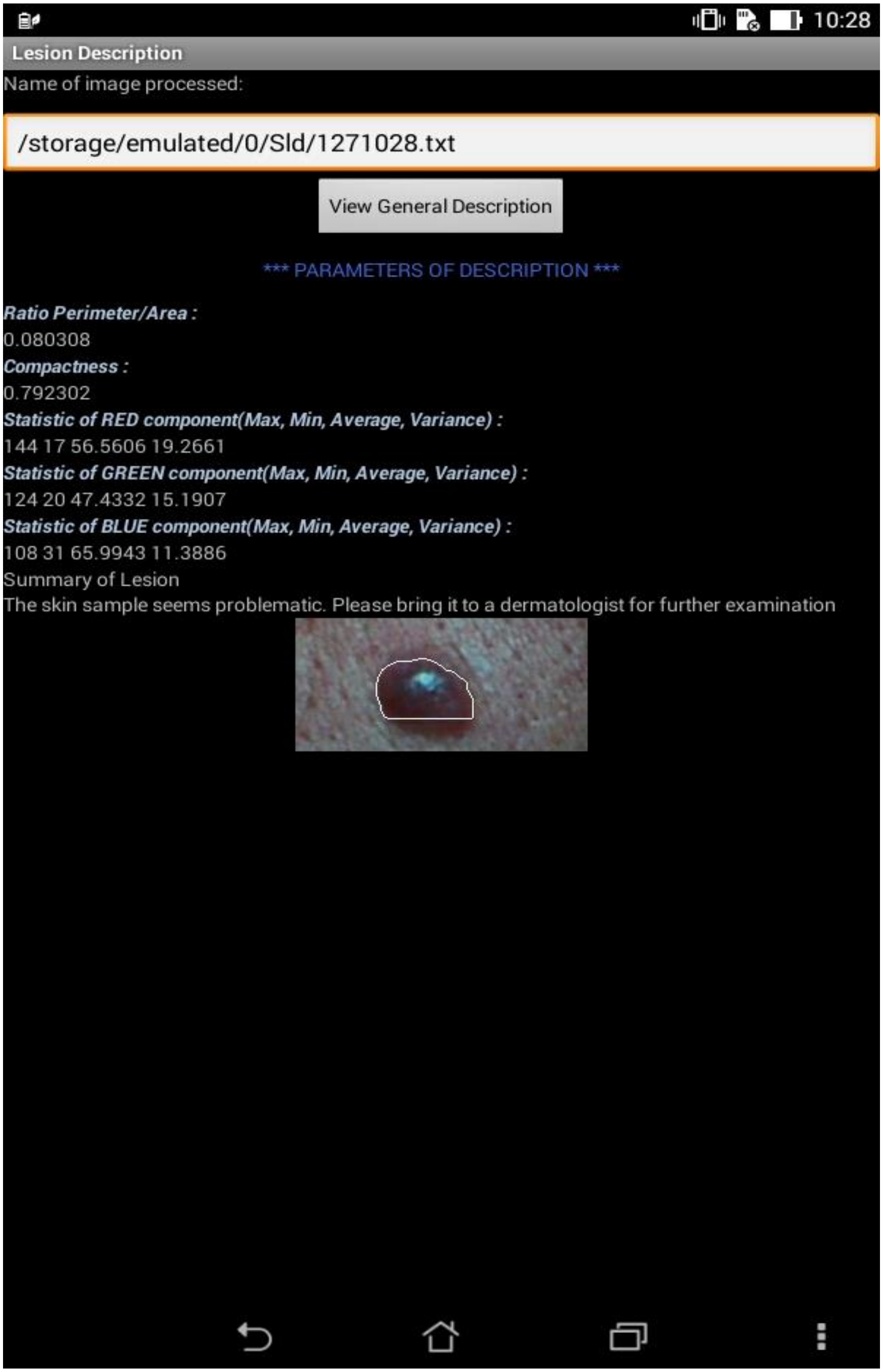


Fig 13c. Lesion analysis and result

Fig 13d. Lesion analysis and result

Fig 13e. Lesion analysis and result

Fig 13f. Lesion analysis and result

Fig 13g. Lesion analysis and result

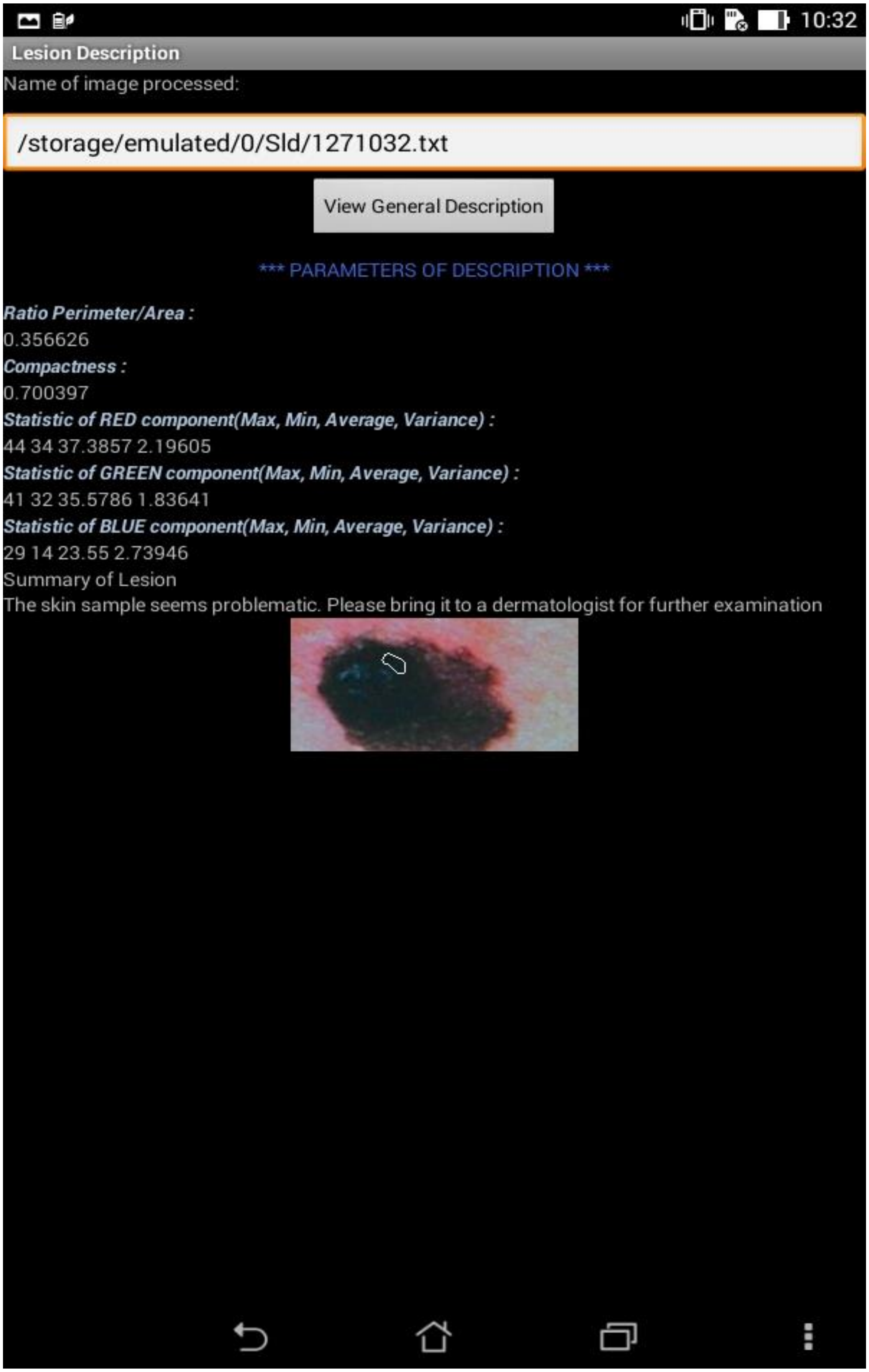


Fig 13h. Lesion analysis and result

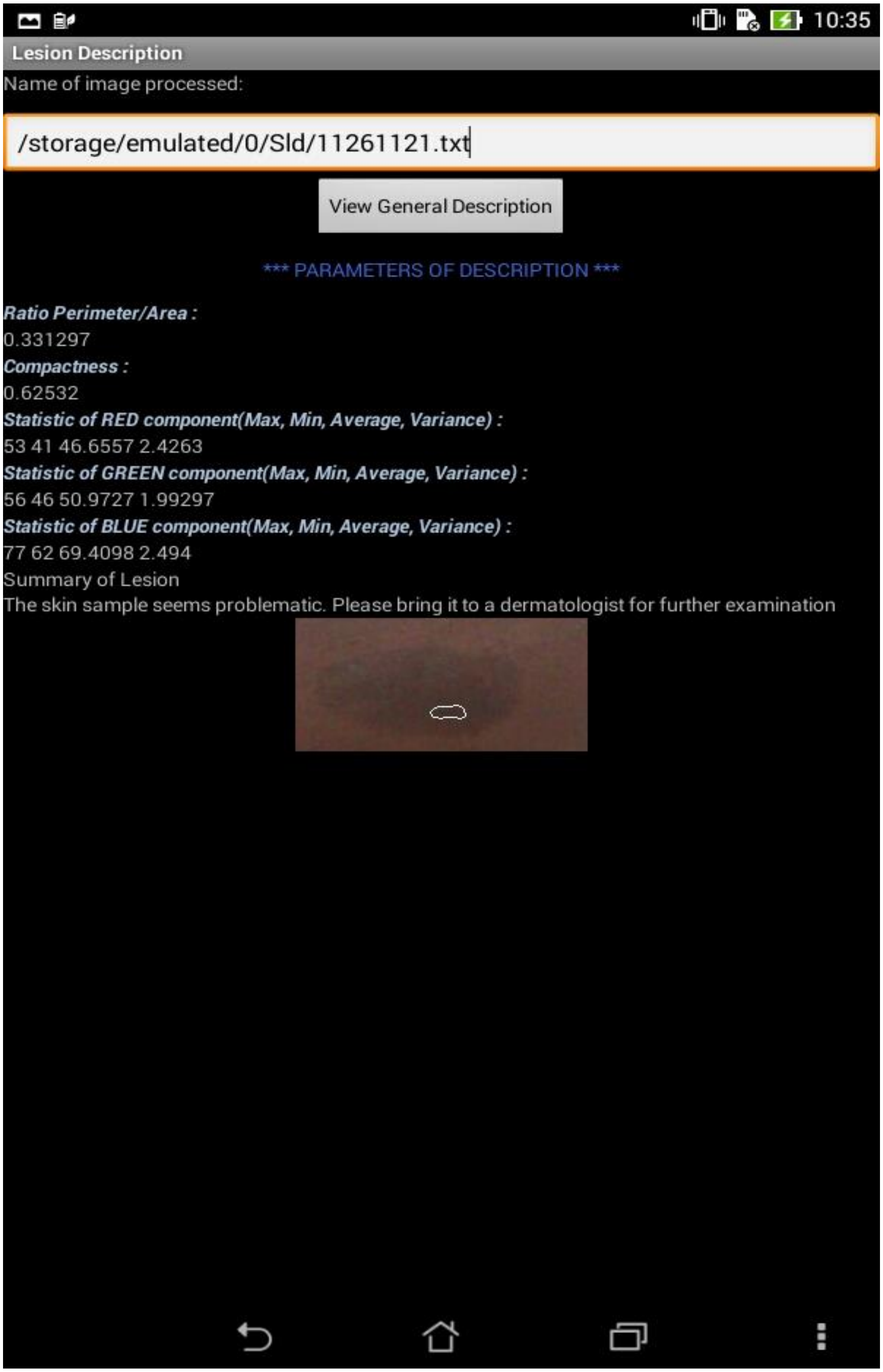


Fig 13i. Lesion analysis and result

Fig 13j. Lesion analysis and result