

# Vision based Pus Segmentation and Area Estimation of Wound using Android Application

Anudeep Varma, Ashok Varma, Hema Madhav Jakkampudi\* and Alex Noel Joseph Raj

School of Electronics Engineering, VIT University, Vellore – 632014, Tamil Nadu, India; anudeepvrm@gmail.com, ashokvitece@gmail.com, jhmadhav@gmail.com, alexnoel@vit.ac.in

## Abstract

The main intention of this paper is to eliminate the use of contact wound measurement techniques such as Ruler method, Graph method, Acetate method etc. These techniques, which are usually inaccurate and cumbersome, can cause pain and inconvenience to the patient. The paper mainly focuses on implementing and comparing various image processing and segmentation algorithms to get accurate measurements of the wounds. The second phase of our project concentrates on implementing these measurement algorithms on an android platform. The wound measurement application allows the patients to capture and assess their wounds from their home or work place. The application can be installed in almost all available smart phones.

**Keywords:** Android, Grab-Cut, K-Means, Non-Contact, Otsu's Method, Wound-Measurement

## 1. Introduction

Digital imaging photography has become quite popular in these days for wound measurement analysis. Traditionally, clinicians performed wound measurement using wound tracing techniques, which involved contact with the wound that could lead to wound contamination risks. The main intention of this paper is to eliminate the use of contact based wound measurement techniques such as Ruler method, Graph method, Acetate method etc. These techniques are usually inaccurate, cumbersome, can cause pain and inconvenience to the patient. This paper mainly focuses on implementing, comparing various image processing and segmentation algorithms to get accurate measurements of the wounds. The measurement parameters that are currently being focused on are wound area, and tissue segmentation (for pus identification)<sup>5</sup>. The project simulates various image processing algorithms on Visual Studio by using OpenCV<sup>29</sup>. OpenCV libraries are preferred over MATLAB in order to improve efficiency, ease of converting the code to android java applications and also to reduce the complexity, so as to run on mobile platforms, which remains to be the ultimate objective

of the project. Final objective is to test the efficiency of various wound processing algorithms and use them to develop an Android application. Many patients across the globe suffer from chronic wounds. These wounds have a very slow and painful healing process, which requires constant monitoring. In many hospitals, contact based wound measurement techniques are used to determine the area and other parameters of the wound. This causes a lot of pain and inconvenience to the patient. Several non-contact wound measurement devices exist in the market that are expensive and require a trained professional to use. In spite of several attempts of developing non-contact wound measurement techniques or devices; we are still in dearth of in-expensive, user-friendly, fast and accurate method. This motivated us to endeavor in developing a portable and a cost efficient non-invasive method for measuring wound dimensions and area.

## 2. Methodology

The project intends to provide suitable algorithms, which can be easily implemented on android applications to assist wound measurement. The main wound measurement

\*Author for correspondence

parameters that are currently being focused are area of the wound, Pus area and Pus percentage. The initial algorithm implementation was done using OpenCV. Upon analyzing the obtained results, suitable algorithms were converted to java android for the app development. The Measurement and analysis process of the project mainly includes 3 steps: Segmentation, Clustering (Pus Identification) and Development of android application.

### 2.1 Segmentation

It is the process of segregating the foreground from the background. In the current project, segmentation mainly focuses on extracting the wound from the image. Accuracy in this step is crucial as this forms the base to various algorithms to follow. Segmentation becomes the pre-processing procedure, which determines the efficiency of various wound measurement parameters. OTSU<sup>17</sup> and Grab-cut<sup>21</sup> segmentation algorithms are used to obtain the wound from the background.

OTSU and Grab-cut segmentation algorithms were used to obtain the wound from the background. Segmentation<sup>24</sup> involves separating an image into regions (or their contours) corresponding to objects. We usually try to segment regions by identifying common properties. Or, similarly, we identify contours by identifying differences between regions (edges). The simplest property that pixels in a region can share is intensity. So, a general way to segment an image is to create a threshold and separate the pixels on comparing with the threshold.

Since the basic threshold includes comparing with a limit, a binary image is produced. All pixels above the threshold are given black and all pixels below are white or vice versa

If  $g(x, y)$  is a threshold version of  $f(x, y)$  at some global threshold  $T$ ,

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{otherwise} \end{cases}$$

OpenCV provides different styles of thresholding and it is decided by the fourth parameter of the function. Different types are:

- cv2.THRESH\_BINARY
- cv2.THRESH\_BINARY\_INV
- cv2.THRESH\_TRUNC
- cv2.THRESH\_TOZERO
- cv2.THRESH\_TOZERO\_INV

An alternative segmentation algorithm used to extract wound is Grab-cut. Most segmentation techniques make

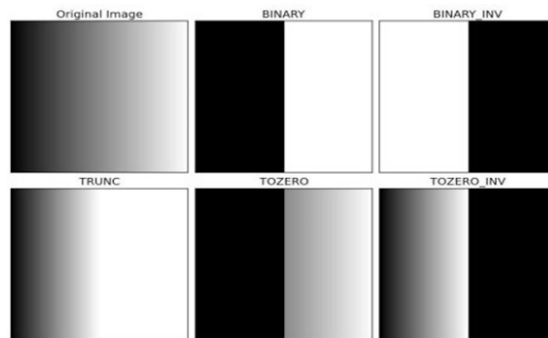


Figure 1. Different styles of thresholding offered by OpenCV.

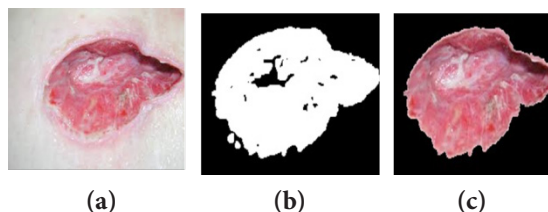


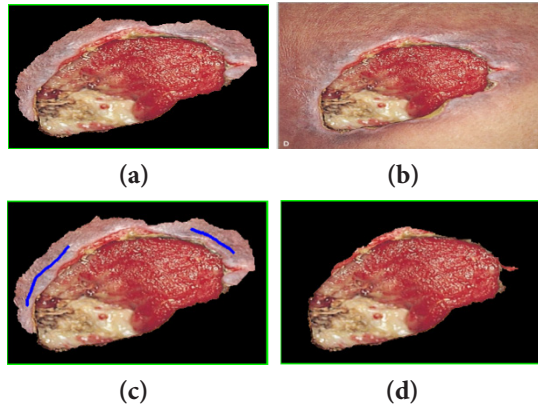
Figure 2. OTSU's Algorithm used for wound segmentation, (a) Diabetic Wound image. (b) OTSU thresholded image of the wound. (c) Segmented wound image.

use of either edge or region information contained in the image in order to perform segmentation. GrabCut<sup>21,34</sup> is an innovative segmentation technique that uses both region and boundary information contained in an image in order to perform segmentation. GrabCut also performs image segmentation in a novel way by using graphs to store region and boundary information. A Min-Cut/Max-Flow algorithm<sup>4</sup>, which is a graph cut technique, is used to segment the graph and in doing so segment the image. Grab Cut also includes a matting technique, which is used to calculate the alpha matte for boundaries of segmented regions.

Otsu's method proved to be fast but was inaccurate with low contrasting images. Hence Grab-cut was used to improve the accuracy at the cost of high computation time. Both the algorithms were implemented in the android application allowing the user to select based on their smart phone's processing capabilities.

### 2.2 Clustering

Main Intention to cluster a wound is to identify and segregate various tissue regions of the wound. Main use of clustering tissues is to identify the wound healing process. The percentage area of various clusters or segments gives the rate and extent of tissue healing. This project mainly



**Figure 3.** Grab cut on a wound image (a) original wound. (b) Grab cut result after 1<sup>st</sup> iteration (without background pixel given as input). (c) Grab cut result after 2<sup>nd</sup> iteration (with background pixel given as input). (d) Grab-cut segmented wound.

focuses on pus segmentation from the wound. Pus usually is a measure of extent of infection in the wound. Extracting the pus percentage can help the doctors analyze the infection and suggest suitable measures.

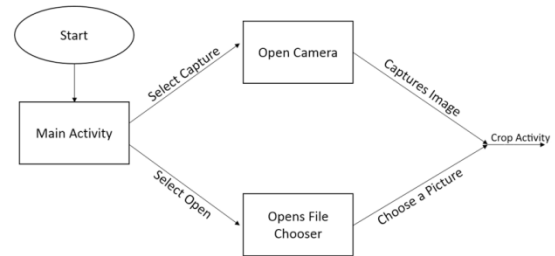
K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining and many other applications. K-means clustering<sup>26</sup> aims to partition  $n$  observations into  $K$  clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

K-means clustering applied to wound image will cluster the image based on the pixel's RGB value. The main aim of the project is to segregate the pus from the wound. In order to achieve that, appropriate  $K$  value has to be fixed so that upon clustering, an entire cluster containing pus pixels is created. Hence a threshold was created to separate the pus cluster from the other clusters. An image database was created and used throughout this project. The above mentioned algorithms were tested on the same image database and their results computed.

### 2.3 Android Application Design

The practical implementation of this project was achieved by the development of an android application. The application was designed to do all the stated algorithms, both segmentation and clustering, for the measurement of wounds.

The application is made of multiple activities that perform above-mentioned algorithms for wound measurement. An Activity<sup>32</sup> is an application component that provides a screen with which users can interact in order to do something, such as dial the phone, take a photo, send an email, or view a map. Each activity is given a window in which to draw its user interface. The window typically fills the screen, but may be smaller than the screen and float on top of other windows.



**Figure 4.** Main activity to crop activity android flow chart.

### 2.4 Home Activity

Activity Home is the first screen which appears on opening the app. It has two buttons from which one can be selected; to open an image that was captured already or to capture an image from the mobile camera.

### 2.5 Using Existing Camera Apps

A quick way to enable taking pictures or videos in an application without a lot of extra code is to use Intent to invoke an existing Android camera application. Camera intent makes a request to capture a picture or video clip through an existing camera app and then returns control back to the application. This section shows how to capture an image or video using this technique.

The procedure<sup>32</sup> for invoking camera intent follows these general steps:

Compose a Camera Intent - Create an Intent that requests an image or video, using one of these intent types:

MediaStore. ACTION\_IMAGE\_CAPTURE - Intent action type for requesting an image from an existing camera application.

MediaStore. ACTION\_VIDEO\_CAPTURE - Intent action type for requesting a video from an existing camera application.

Start the Camera Intent - Use the startActivityForResult() method to execute the camera intent. After intent is started, the Camera application user interface appears on

the device screen from which a picture or video can be taken.

Receive the Intent Result - Set up an onActivityResult() method in the application to receive the callback and data from the camera intent. The system calls this method, when taking a picture or video is finished (or cancels the operation).

## 2.6 Crop Activity

This Activity is used to crop the Image. The settings give ability to use the default crop activities, which comes with editing apps in android or the custom crop activity, which was embedded in the app. The custom crop activity gives features like crop, rotate and to check the cropped image. Once the crop was executed perfectly, the cropped bitmap can be passed to the next activity using a button.

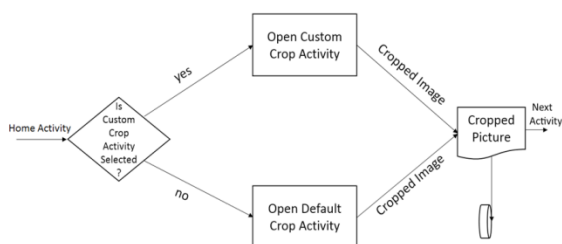


Figure 5. Crop to k-mean android activity.

## 2.7 Segmented Image Activity

In this activity OpenCV4Android will be initialized at first and this activity will access the cropped image the storage and convert it to bitmap. The bitmap is then converted to OpenCV mat object so the image can be processed in OpenCV.

Based on settings the image is passed to either Grab Cut or Otsu for separating skin pixels from image. The Otsu thresholding function works similar to the one implemented in visual studio. But Grab Cut function was implemented differently because of hardware restrictions, background pixels can't be sent to the function directly as in the visual studio, so instead of sending background pixels to the function increase in the number of iterations can help. On every iteration, the background pixels get added automatically. But this method comes with a cost. As number of iterations increase, the Grab Cut method takes longer time to process the image. Hence a tradeoff had to be made between accuracy and processing time.

One can chose between accuracy and fast response by switching between Grab Cut and Otsu.

Once the skin pixels are separated from the Image, it is then sent to the K-mean function. This function segments the image into clusters based on number of clusters selected. Higher the number of clusters, closer the clustered image resembles the original one. Lesser clusters mean that the pus can be segmented as one cluster. So the optimum number of clusters was found to be 3. On segmenting the wound region into three clusters they are defined based on their mean color. The intermediated images in these three functions are stored in bitmaps. When the appropriate buttons are clicked, respective images will be displayed. To save resources from jamming, the images are stored in bitmap objects in the ram. So they are easy and quite fast to access. To prevent app from flooding the memory, intermediate objects created will be destroyed and memory will be freed once the app is closed.

Once these segmentations are done, the activity displays the original image and various buttons (options) to view any one of the four processed images. Now the final image will be sent to the "calculate" function which computes the area of the image.

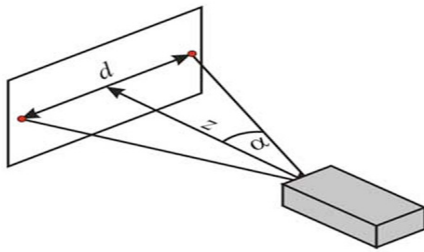
## 2.8 Area Calculation (Android Application)

The next major step in application development was the challenge of converting digital pixel into real world measurement (cm<sup>2</sup>).

Various conversion techniques exist which uses lasers and scales. A simpler concept was implemented in the application, which only requires the user to know the distance between the camera and the wound. The proposed idea would use the view angle of the camera lens from any height x. The application, on knowing the value of x, calculates the area in cm<sup>2</sup> using the concepts of trigonometry. Each camera lens has different horizontal and vertical viewing angles. First, the way in which the application obtains these angles is explained followed by the mathematical computation of the area using these angles.

The length and breadth of the image are calculated using both the angles and height, which the distance from the camera to the object is. From the image,  $\alpha$  will be half of horizontal view angle. Once  $\alpha$  is calculated and z is known, which is the distance between camera and the object, it is easy to find the values of horizontal length of the image using below formula.





**Figure 6.** Calculation of pixel area.

$$v.l = 2(h) \tan (\beta / 2)$$

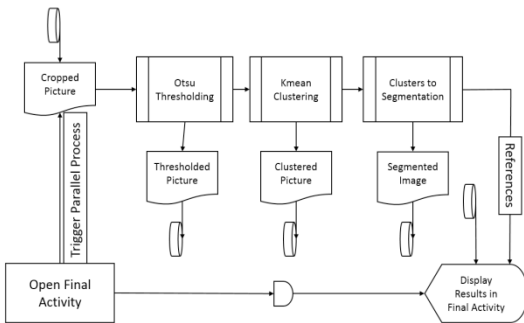
Where

$\beta$  is horizontal view angle.

$\alpha$  is vertical view angle.

$h$  is height i.e. distance of object from camera module.

$v.l$  and  $h.l$  are vertical and horizontal lengths of the image.



**Figure 7.** Complete flow chart of android application.

### 3. Results and Analysis

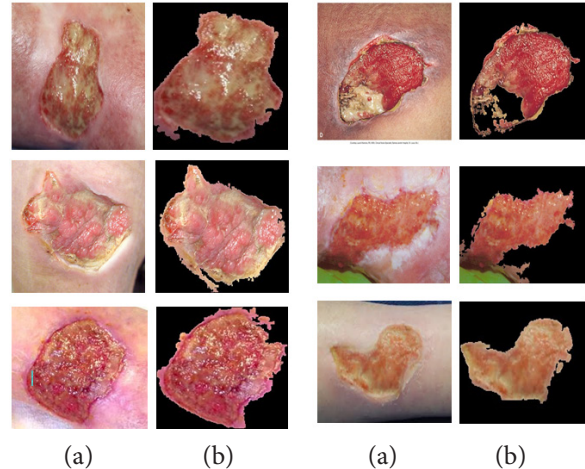
#### 3.1 OTSU’s Results

Visual Studio 2012 was the IDE used to write the codes for various algorithms. OpenCV library was integrated into the IDE to perform image processing on various wound images. An image database of 30 wounds was obtained and OTSU’s algorithm was tested on these images to give suitable results.

The wound areas were calculated using OpenCV. The non-black pixels represent the approximate wound pixels. The actual number of pixels of wound is obtained by using a cropping tool to manually crop the wound out of the image.

#### 3.2 Grab-Cut Results

Grab-Cut Algorithm is an iterative process, whose accuracy increases with the number of iterations. To



**Figure 8.** OTSU’s method results. (a) Wound image. (b) OTSU’s threshold image.

**Table 1.** OTSU’s method area error percentage (All data in pixels)

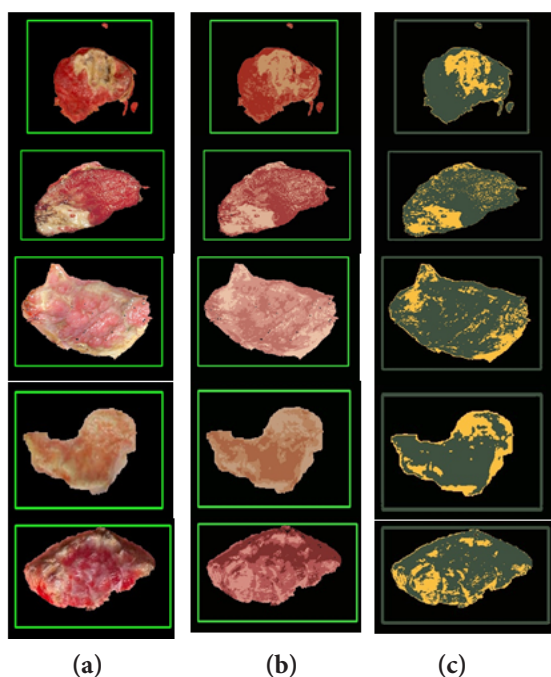
Fig No.	Actual Area	Otsu	Otsu error	Otsu error percent (%)
1	12245	4478	7767	63.42997142
2	16011	18089	2078	12.97857723
3	18496	15627	2869	15.51146194
4	57339	60041	2702	4.712324945
5	36325	34097	2228	6.133516862
6	33832	38852	5020	14.83802317
7	17836	14776	3060	17.15631307
8	4556	4226	330	7.243195786
9	19892	24661	4769	23.9744621
10	8315	15505	7190	86.47023452
11	25093	44853	19760	78.74706093
12	6262	3932	2330	37.20855957
13	84452	143010	58558	69.338796
14	18421	37467	19046	103.3928668
15	42108	65837	23729	56.35271207
16	54702	37808	16894	30.88369712
17	3148	9858	6710	213.1512071
18	34194	41104	6910	20.20822366

compare the results obtained by both the segmentation algorithms, wound area was calculated using OpenCV.

The comparison graph between the two algorithms shows that high contrast images produce similar segmentation. However grab-cut algorithm works efficiently with low contrasting images while the OTSU’s method fails miserably with the same.

### 3.3 K-Mean Clustering Results

K-mean clustering principle is applied to the segmented wound to detect pus as one of its cluster. After many trails on the wound database,  $K = 3$  is found to produce most accurate pus clustering. The images below are the pus-segmented images from the database. The Grab-cut segmented image was K-means clustered to differentiate various regions (number of regions depends on value  $k$ ). The pixel count of the Pus cluster was tabulated and percentage of the pus was calculated below.



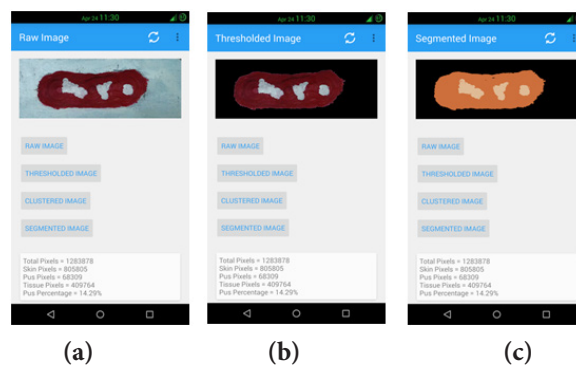
**Figure 9.** K-mean results. (a) Grab-cut image. (b) K-means clustered image. (c) Pus identified region (yellow).

### 3.4 Android Application Demonstration

Successful functioning of the wound measurement android application holds the key to the success of the project. It represents the practical application of the project and distinguishes the current work from the previous wound measurement projects.

The above Figure shows stepwise activities from capture to crop.

The measured parameters, such as wound area and the pus area, are displayed below the images. The measured data is displayed both in pixels and in  $\text{cm}^2$ . The application is programmed in such a way that it is able to calculate the pixel area in  $\text{cm}^2$  irrespective of the height from which the image is captured.



**Figure 10.** Processed images of the wound displayed on the android application along with the wound and pus areas (in pixels). (a) The raw image of the wound. (b) The thresholded image of the wound (segmented the wound from the background). (c) The pus identified region of the wound.

## 4. Conclusion

The android application successfully implements the wound measurement algorithms that were tested using Visual Studio with OpenCV. The segmentation algorithms, namely OTSU and Grab-Cut, extract the targeted wound from the background with as minimum human interaction as possible. However results show that Grab-cut performs better than OTSU's method while segmenting wounds. The segmented wound is then clustered using K-Means clustering to segregate various tissue regions. The project mainly aims at identification of the pus region and measurement of pus percentage. The K-means algorithm clusters the wound into 3 regions ( $K = 3$ ) with the lowest intensity region being pus. Upon measuring the region of pus, the pus percentage will be calculated and corresponding results displayed on the android application. All the measured values that are obtained in pixels can be converted into  $\text{cm}$  by using the field view angle of the camera. Current work can be extended to measure the volume of the wound using a 3D camera. The output of this project enables the patients to analyze their wounds from the comfort of their rooms, with added ease and simplicity. This in turn can reduce the burden on health care centers.

## 5. References

1. Lazarus GS, et al. Definitions and guidelines for assessment of wounds and evaluation of healing. *Wound Repair and Regeneration*. 1994; 2(3):165-70.

2. Nemeth ME, Sprigle S, Gajjala A. Clinical usability of a wound measurement device. 36th Annual American Spinal Injury Conference; Nashville, TN. 2010.
3. Sundeep Kumar K, Eswara Reddy B. Digital analysis of changes in chronic wounds through image processing. *International Journal of Signal Processing, Image Processing and Pattern Recognition*. 2013; 6(5).
4. Marsh M. Grab Cut - Interactive foreground extraction using iterated graph cuts. 2015 Mar. Available from: <http://www.cs.ru.ac.za>
5. Mukherjee R, Manohar DD, Das DK, Achar A, Mitra A, Chakraborty C. Automated tissue classification framework for reproducible chronic wound assessment. *BioMed Research International*. 2014.
6. Bradski G, Kaehler A. O'Reilly Learning OpenCV.
7. Chang AC, Dearman B, Greenwood JE. A comparison of wound area measurement techniques: Visitrak versus photography. *Eplasty*. 2011; 11.
8. Bhat SS, Smith DJ. Laser and sound scanner for non-contact 3D volume measurement and surface texture analysis. *Physiological Measurement*. 1994; 15(1):79.
9. Patete PV, et al. A non-invasive, three-dimensional, diagnostic laser imaging system for accurate wound analysis. *Physiological Measurement*. 1996; 17(2):71.
10. Plassmann P, Jones BF, Ring EFJ. A structured light system for measuring wounds. *The Photogrammetric Record*. 1995; 15(86):197-204.
11. Shaw J, Bell PM. Wound measurement in diabetic foot ulceration. *Global Perspective on Diabetic Foot Ulcerations*. INTECH Open Access Publisher; 2011.
12. Shaw J, et al. An evaluation of three wound measurement techniques in diabetic foot wounds. *Diabetes Care*. 2007; 30(10):2641-2.
13. Van Poucke S, et al. Comparative analysis of two methods for wound bed area measurement. *International Wound Journal*. 2010; 7(5):366-77.
14. Houghton PE, et al. Photographic assessment of the appearance of chronic pressure and leg ulcers. *Ostomy Wound Management*. 2000; 46(4):20-35.
15. Duckworth M, et al. A clinically affordable non-contact wound measurement device. RESNA Annual Meeting; Phoenix, AZ. 2007.
16. Hettiarachchi NDJ, et al. Mobile based wound measurement. *IEEE Point-of-Care Healthcare Technologies (PHT)*. 2013.
17. Otsu N. A threshold selection method from gray-level histograms. *Automatica*. 1975; 11(285-296):23-7.
18. Tolga Birdal. The basics of the famous Otsu's automatic thresholding method with an implementation in C#. 2015 Feb 21. Available from: <http://www.codeproject.com/Articles/38319/Famous-Otsu-Thresholding-in-C>
19. Gonzalez RC. *Digital Image Processing*.
20. OpenCV datasheet. 2015 Jan 12. Available from: [http://docs.opencv.org/master/d7/d4d/tutorial\\_py\\_thresholding.html](http://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html)
21. Rother C, Kolmogorov V, Blake A. GrabCut - Interactive foreground extraction using iterated graph cuts. *Microsoft Research Cambridge-UK*.
22. Yuri B, Marie-Pierre J. Interactive graph cuts for optimal boundary and region segmentation of objects in ND images. *IEEE International Conference on Computer Vision*; USA. 2001.
23. Justin F. Talbot XX. *Implementing Grab Cut*. 2006.
24. Hegadi RS, Goudannavar BA. Interactive segmentation of medical images using grabcut. *International Journal of Machine Intelligence*. 2011; 3(3).
25. Verma RM, Chen P. A data mining hypertextbook: Design, implementation and experience. *Journal of Computing Sciences in Colleges*. 2012; 27(3):22-8.
26. Hartigan JA, Wong MA. Algorithm AS 136: A k-means clustering algorithm. *Applied Statistics*. 1979:100-8.
27. Weeks AR, Jr. *Fundamentals of Electronic Image Processing*. IEEE Press; 1996.
28. Color models. 2004 Mar 10. Available from: <http://academic.mu.edu/phys/matthysd/web226/index.htm>
29. About OpenCV. 2015 Jan 15. Available from: <http://opencv.org>
30. Matlavs OpenCV. Available from: <http://blog.fixational.com/post/19177752599/opencv-vs-matlab>, <http://www.aishack.in/2010/02/why-opencv/>, <http://stackoverflow.com>
31. Basics of Image processing. 2015 Jan 7. Available from: <http://www.engineersgarage.com>
32. Android datasheet. 2015 Mar 10. Available from: <http://developer.android.com>
33. Conversion of pixel area to cm<sup>2</sup>. 2015 Mar 28. Available from: <http://imageprocessingblog.com/image-segmentation-techniques-for-tracking-fish-wound>
34. Rother C, Kolmogorov V, Blake A. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*. 2004; 23(3):309-14.
35. Developing OpenCV for Android. 2015 Mar 13. Available from: <http://www.embedded.com/design/programming-languages-and-tools>
36. Plassmann P, Jones TD. MAVIS: A non-invasive instrument to measure area and volume of wounds. *Medical Engineering and Physics*. 1998; 20(5):332-8.
37. Gaur A, Sunkara R, Raj ANJ, Celik T. Efficient wound measurements using RGB and depth images. *International Journal of Biomedical Engineering and Technology*. 2015; 18(4).