2007

Detection of flash in dermoscopy skin lesion images

Radhika Nagane

DETECTION OF FLASH IN DERMOSCOPY SKIN LESION IMAGES

by

RADHIKA NAGANE

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2007

Approved by

R. H. Moss, Advisor

R. J. Stanley

W.V. Stoecker

ABSTRACT

Dermoscopy images of skin lesions are often acquired by covering the skin with mineral oil, alcohol, another liquid or a gel and illuminating the area at a low angle of incidence from all directions before taking the photograph. Deeper pigment patterns are more visible in this type of image than in a clinical image because the liquid immersion and structured lighting effectively make the upper layers of the skin more transparent. Flash regions are the small, high reflectance areas in the image where direct reflections of the illumination source(s) cause undesirable bright spots in the image. These flash regions may be due to reflections from liquid trapped in the rough textured areas of the lesion or to reflections from bubbles in the liquid.

This research focuses on the development and evaluation of algorithms to detect and remove high reflectance regions characteristic of flash. Techniques are also explored to color correct the image in detected high reflectance regions to make the image look natural. After flash removal, it becomes easier to extract the features of the image more accurately.

ACKNOWLEDGMENTS

I would like to express my deep and sincere gratitude towards my advisor, Dr. Randy Moss, whose knowledge and logical way of thinking have been of great value for me. His understanding, encouraging and personal guidance have provided a good basis for the present thesis. I would like to thank my committee members, Dr. Joe Stanley and Dr. W. V. Stoecker, for their detailed and constructive comments and for their support throughout my research work.

During this work, I have collaborated with many colleagues for whom I have great regard. I wish to thank Austin Bangert for all his efforts in drawing the lesion borders. I would also like to extend my thanks to Kapil Gupta and Shoba Umamaheswaran, whose previous work was of great help in my research.

Finally, I owe my most sincere gratitude towards my parents, brother and sister-in-law for their continued support and blessing towards achieving this goal.

TABLE OF CONTENTS

Page

ABSTRACT iii

ACKNOWLEDGMENTS iv

LIST OF FIGURES viii

LIST OF TABLES ix

SECTION

1. INTRODUCTION 1

2. THE FLASH REMOVAL ALGORITHM 3

2.1 OVERVIEW 3

2.2 DETAILS OF THE STEPS INVOLVED 5

2.2.1. Pre-processing 5

2.2.1.1. Border removal 5

2.2.1.2. Hair removal. 6

2.2.2. Border Mask Generation 7

2.2.3. Relative Color Image Generation 8

2.2.4. Finding Flash Mask 8

2.2.4.1. Threshold calculation by adding a fixed number 9

2.2.4.2. Threshold calculation by adding a constant times the Standard deviation 11

2.2.4.3. Threshold calculation using curve-fitting equation....................13

2.2.5. Color-smoothing Algorithm (Region Filling)...............................14

3. GRAPHICAL USER INTERFACE (GUI)...15

4. RESULTS...20

4.1 EXPERIMENTS PERFORMED 20

4.2 SCORING OF RESULTS...21

4.2.1. Pixel-based Method. 21

4.2.2. Region-based Method...22

5. CONCLUSION...33

APPENDICES

A. MAIN PROGRAM FOR FLASH FINDING (FinalFlash.m) 34

B. CALCULATING THE RELATIVE COLOR IMAGE (RelColCalculate.m)......36

C. SUB-ROUTINE FOR FINDING FLASH MASK (FindFlash.m)......................40

D. SUBROUTINE TO REMOVE THE LARGE FLASH AREAS DETECTED MISTAKENLY (RemLargeFlash.m) 46

E. SUB-ROUTINE FOR FILLING THE FLASH REGION (FillFlash.m)..............48

F. SCORING RESULTS OF FLASH REMOVAL ON TRAINING SET OF 85 IMAGES 57

G. MULTIPLIER REDUCTION ALGORITHM BY CHOWDHURY-WRONKIEWICZ...69

H. SCORING RESULTS OF FLASH REMOVAL ON TEST SET OF 99 IMAGES............73

BIBLIOGRAPHY............89

VITA............91

LIST OF FIGURES

Figure Page

2.1 Flowchart of the flash removal algorithm....................4

2.2 Image example with border removed....................6

2.3 Image example from Figure 2.2 (b) with hair removed using SharpRazor technique...7

2.4 Lesion mask for lesion from Figure 2.2 (b).................... ...7

2.5 Flash mask found by Fixed Number Method....................10

2.6 Flash mask found with loose threshold combined with salient feature mask....................11

2.7 Examples of flash mask for Standard Deviation Method....................13

3.1 Layout of the GUI for flash removal....................15

3.2 Mode and process selection for GUI... 17

3.3 Image selection for the GUI....................17

3.4 Process bar indicating the progress of image processing....................18

3.5 Final view after the processing is complete for single mode....................19

4.1 Curve fitting equation for Green plane of test set images using MATLAB® curve fitting tool....................23

4.2 Example Image (Abl108.tif) from training set....................26

4.3 Example of flash removal for Fixed Number Method....................27

4.4 Example of flash removal for Standard Deviation Method 27

4.5 Example of flash removal for Curve Fitting Method....................27

4.6 Example of the Curve Fitting Method finding excessive flash regions....................28

4.7 Another example of the Curve-Fitting Method finding excessive flash regions....................28

4.8 Example of image with missed flash regions....................29

4.9 Image example with flash removal and color correction successfully performed using the Curve Fitting Method....29

4.10 Image example with flash removal and color correction successfully performed using the Standard Deviation Method....30

4.11 Another image example with flash removal and color correction successfully performed using the Curve Fitting Method....30

LIST OF TABLES

Table Page

4.1 Summary table for results of scoring for the 85 images in the training set................22

4.2 Summary table for results of scoring for the 99 images in the test set........................24

4.3 Summary of results for Fixed Number Method...24

4.4 Summary of results for Fixed Number Method combined with Salient feature masks...25

4.5 Summary of results for Standard Deviation Method...25

4.6 Summary of results for Standard Deviation Method combined with Salient feature masks...26

4.7 Summary of results for Curve Fitting Method..32

1. INTRODUCTION

Malignant melanoma is the most deadly form of skin cancer with 59,940 new cases and 7,110 deaths estimated in the United States in 2007 [1]. If not detected at an early stage, it can be fatal. Dermoscopy, also called dermatoscopy, epiluminescence microscopy or skin surface microscopy, is a non-invasive technique allowing magnified examination of skin lesions to provide improved clinical diagnosis of malignant melanoma. High-reflectance bright areas, which are called flash areas here, can cause difficulties in automated feature detection and can be confusing or at least annoying to humans analyzing the image.

This research focuses on removing flash to provide improved visibility of the image. The idea of flash removal came up during the development of software to detect white areas in lesions. In a few images the flash areas were getting detected as white areas due to their brightness. Also, flash creates a problem in the feature extraction of the image which may lead to a false diagnosis. The aim of this research is to process the images for flash removal and produce images with better visibility.

A set of 85 dermoscopy images was used as a training set for algorithm development. A disjoint test set of 11 dermoscopy images was used for algorithm evaluation. The initial part of this thesis describes the pre-processing steps involved before flash removal, such as border removal and hair removal. It is followed by the algorithm for the flash removal procedure and a detailed explanation of each step. Different features of the image were extracted and analyzed to determine the appropriate threshold for finding the flash mask. Based on the empirical analysis of the training set of images, various methods were developed for flash finding. These flash areas were then filled using a color smoothing algorithm to give the final flash-removed image. A Graphical User Interface (GUI) was developed to make the procedure easy. The examples and results are summarized at the end in Section 4. Two methods were used for scoring the results, the pixel-based method and the region-based method. The results are summarized for these methods followed by the conclusion and future scope.

2. THE FLASH REMOVAL ALGORITHM

2.1 OVERVIEW

The steps in the flash removal process for the different algorithms investigated in this research are as follows:

1. Preprocessing
    - Border removal
    - Hair removal
2. Generating lesion masks
3. Generating relative color image
4. Finding flash mask
    - Fixed threshold method: With threshold calculated by adding a constant to the Otsu threshold.
    - Standard deviation method: With threshold calculated by adding a constant times the standard deviation of the lesion to the Otsu threshold
    - Curve fit method: With threshold calculated using curve fitting equation.
5. Region filling

The flowchart for the entire process is shown in Figure 2.1.

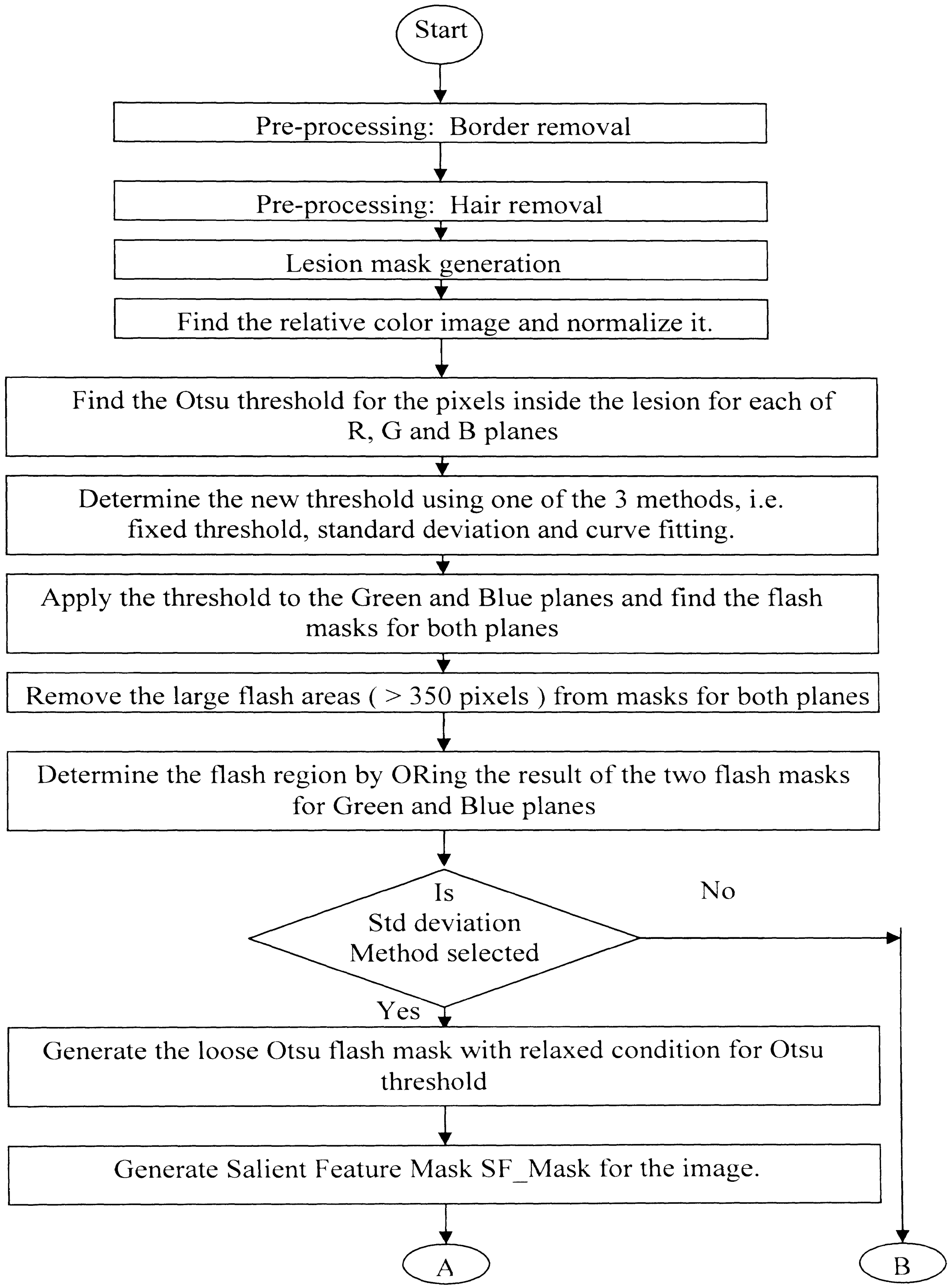


Figure 2.1 Flowchart of the flash removal algorithm

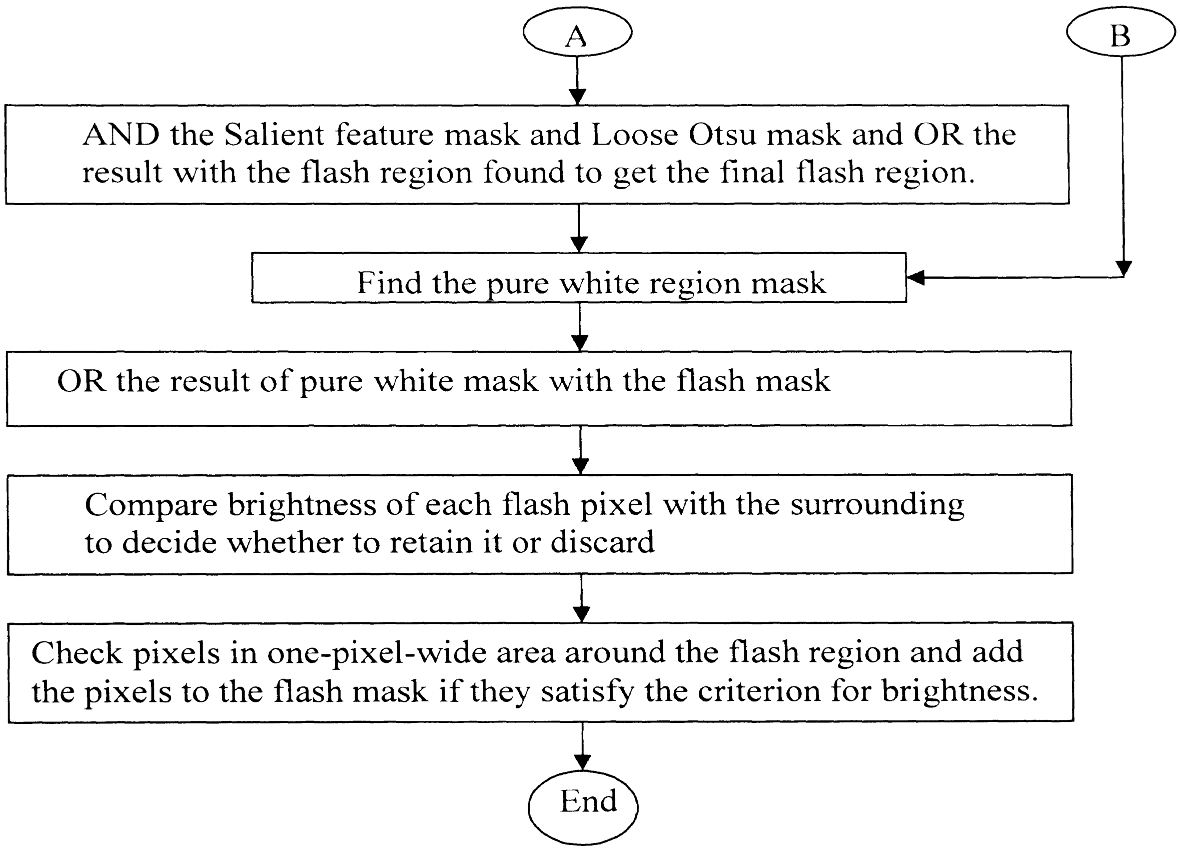


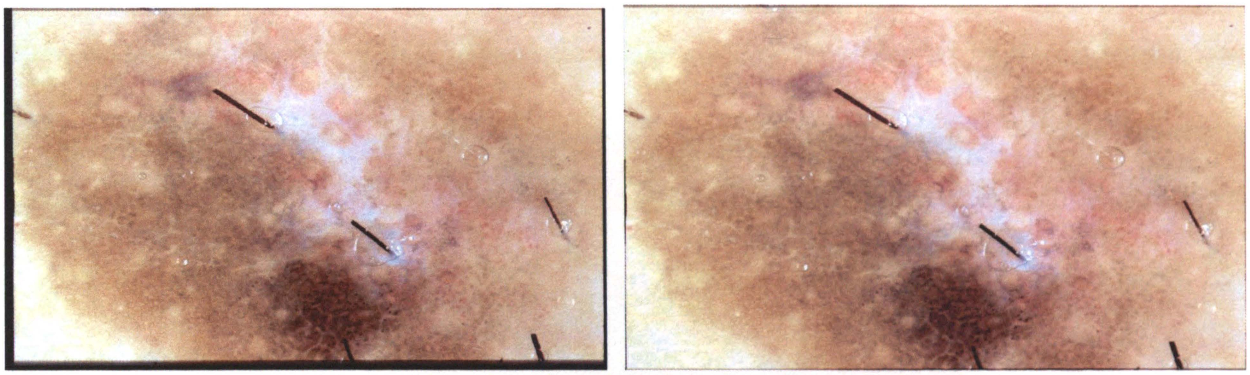
Figure 2.1 Flowchart of the flash removal algorithm (cont.)

2.2 DETAILS OF THE STEPS INVOLVED

The steps mentioned in Section 2.1 are explained in detail as follows.

2.2.1. Pre-processing. Some of the images received from clinics have black borders (perhaps from slide digitization), which may interfere while analyzing the image features. Sometimes the lesion is covered with hair, which hides the details of the image. Preprocessing is required in order to remove such unwanted objects and make it easier to analyze the image. The steps involved in preprocessing are as follows.

2.2.1.1. Border removal. Some dermoscopy images have a dark black border around them as shown in Figure 2.2(a). Sometimes flash areas are present near the border of the image. In such cases while filling the flash areas with the region-filling technique (described later in Section 2.2.5) the black color is used as a fill color which is extremely undesirable. To avoid this, black border removal [2] is done as a preprocessing step. Figure 2.1 shows an original image with a black border and the border-removed image.



(a) (b)

Figure 2.2 Image example with border removed

(a) Original image, (b) Border-removed image

2.2.1.2. Hair removal. Before removing the flash region, the image needs to be pre-processed, to get rid of non-skin or foreign objects like hairs and bubbles. (Sometimes air can be trapped between the skin and the camera lens in the liquid layer to produce bubbles.) To remove the hair from the images, a program called SharpRazor, an improved version of DullRazor [2, 3, 4], is used. SharpRazor performs the following steps to remove the hair:

i. It identifies dark hair locations by a generalized grayscale morphological closing operation.

ii. It verifies the shape of the hair as a thin and long structure and replaces the verified pixels using bilinear interpolation.

iii. It smoothes the replaced hair pixels using the 'roifill' function in MATLAB® which smoothly interpolates inward from the pixel values on the boundary of the polygon by solving Laplace's equation.

SharpRazor detects and removes most of the hair in the images; though traces of hair remain in some cases. It also removes some of the bubbles partially or fully. Figure 2.3 shows the results of applying SharpRazor to the image in Figure 2.2 (b).

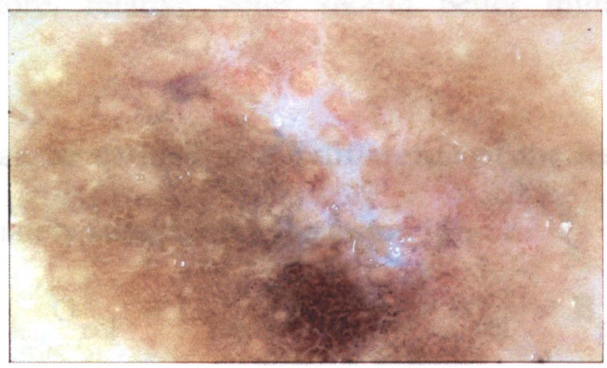


Figure 2.3 Image example from Figure 2.2 (b) with hair removed using SharpRazor technique

2.2.2. Border Mask Generation. Lesion borders were drawn manually using the software WinShow (which creates border masks), and all the borders were checked by a dermatologist. Figure 2.4 shows the lesion mask for the image of Figure 2.2 (b).

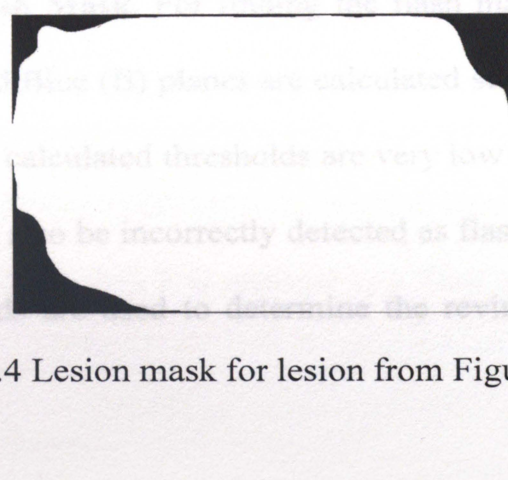


Figure 2.4 Lesion mask for lesion from Figure 2.2 (b)

2.2.3. Relative Color Image Generation. Relative color information has been used successfully to process clinical and dermoscopy images [5, 6, 7, 8, 9, 10]. The relative color concept is useful for the following reasons [11]:

1. It tends to equalize any variations caused by lighting, photography/printing or the digitization process.
2. It tends to equalize variations in normal skin color between individuals.
3. The human visual system works on a relative color system as opposed to the absolute color system. Therefore, relative color should better mimic the dermatologist's diagnostic process.

Relative color can be defined as the difference of the lesion pixel color and the color of the skin immediately surrounding the lesion [6], i.e.

$$R_{rel} = R_{pix} - R_{avg} \; ; \quad G_{rel} = G_{pix} - G_{avg} \; ; \quad B_{rel} = B_{pix} - B_{avg}$$

where, Ravg, Gavg and Bavg are mean red, green and blue respectively for the 20-pixel-wide area around the lesion boundary (which basically is the surrounding skin area; hence the mean shows the average skin color) and R_{pix}, G_{pix} and B_{pix} are the red, green, and blue values of a pixel within the lesion. R_{rel}, G_{rel} and B_{rel} are the values of the relative color for that pixel.

2.2.4. Finding Flash Mask. For finding the flash mask, the Otsu thresholds for the Red (R), Green (G) and Blue (B) planes are calculated separately for the pixels inside the lesion area [12]. These calculated thresholds are very low and if they are used directly then less bright pixels will also be incorrectly detected as flash regions. In order to avoid that, the following methods are used to determine the revised threshold which will be higher than the actual Otsu value. Also, only the G and B planes are used for finding the flash masks as it was observed that the use of the R plane tends to result in too many false positive flash areas being found.

2.2.4.1. Threshold calculation by adding a fixed number. In this method, based on the value of the Otsu threshold, a constant number is added to the calculated Otsu threshold. These fixed values are determined empirically. The conditions are as follows:

Let,

R_{cal} = calculated Otsu threshold for Red plane

G_{cal} = calculated Otsu threshold for Green plane

B_{cal} = calculated Otsu threshold for Blue plane

If $B_{cal} < 0.5$ then add 0.25 to B_{cal} and G_{cal}

If $G_{cal} > 0.5$ and $B_{cal} > 0.5$ then use G_{cal} and B_{cal}

If $R_{cal} > 0.85$ and $G_{cal} > 0.5$ and $B_{cal} > 0.5$ then add 0.1 to G_{cal} and B_{cal}

The flash masks are found separately for the Green and Blue planes using the thresholds found using the above conditions, and the results are ORed after removing large flash areas (areas > 350 pixels), as sometimes bright areas of surrounding skin color are falsely detected.

The next step is verification, in which for each flash pixel found in the above mask, a 40 x 40 window is considered around that flash pixel (for each of the R, G and B planes). If the candidate flashes pixel value is brighter than the average of the window for any of the R, G or B planes, then it is retained as a flash pixel in the mask found; else it is discarded.

The final step is flash expansion. In order to capture the somewhat less bright flash pixels just around the flash area as well as to avoid smearing nearby duller flash pixels into the filled areas (so that the color filling will be smoother), a one-pixel wide area around the flash border is considered. For each pixel in this area, if the R, G, and B values for that pixel are all greater than 200, then the pixel is added to the flash mask.

An example of the flash mask for the image shown in figure 2.2(b) using this technique is shown in Figure 2.5.

Figure 2.5 Flash mask found by Fixed Number Method

In some cases, this algorithm misses some of the obvious flash areas. In order to overcome this problem, a salient feature mask [13] is combined with the flash mask. A flash mask is found using somewhat loose threshold conditions to the G and B planes to capture the less bright flash region and then it is ANDed with the salient feature mask (found using the salient feature code [13]). The resultant mask is ORed with the original flash mask (i.e. the flash mask found with original threshold, not the loose threshold) to get more accurate results.

The loose (relaxed) threshold conditions for the G and B planes are:

If $B_{cal} < 0.5$ then add 0.24 to B_{cal} and G_{cal}

If $G_{cal} > 0.5$ and $B_{cal} > 0.5$ then use G_{cal} -0.09 and B_{cal} -0.09

If $R_{cal} > 0.85$ and $G_{cal} > 0.5$ and $B_{cal} > 0.5$ then add 0.09 to G_{cal} and B_{cal}

An example of a flash mask with a relaxed threshold combined with a salient feature mask is shown in Figure 2.6.

Figure 2.6 Flash mask found with loose threshold combined with salient feature mask

2.2.4.2. Threshold calculation by adding a constant times the standard deviation. In this method, instead of adding a constant to the calculated Otsu threshold, a fraction of the standard deviation of a color plane of the area inside the lesion (determined empirically as 0.25 times the standard deviation) is added to the Otsu threshold. The thresholds are calculated as follows:

- Add 0.25 times the standard deviation of the green plane for the lesion area to G_{cal}.

- Add 0.25 times the standard deviation of the blue plane for the lesion area to B_{cal}.

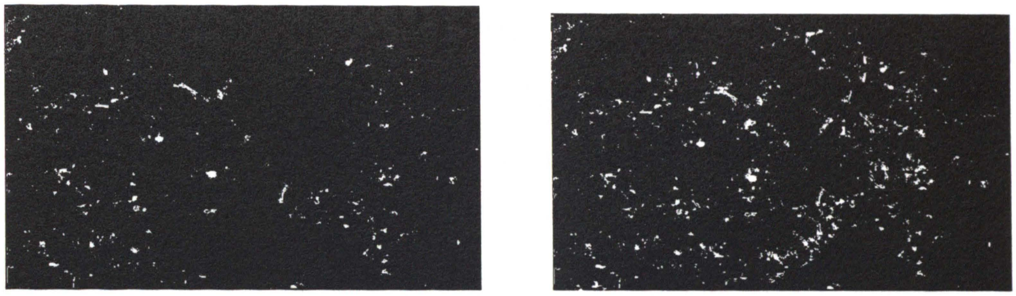
The flash masks for the G and B planes are found using the thresholds calculated using the above conditions, and an OR operation is performed on the resulting binary masks to obtain the preliminary flash mask.

In order to find some flash areas which would otherwise be missed, a relaxed Otsu threshold is calculated as follows.

- Add 0.2 times the standard deviation of the green plane for lesion area to G_{cal}.
- Add 0.2 times the standard deviation of the Blue plane for lesion area to B_{cal}.

Using these thresholds, a relaxed flash mask is found by ORing the mask results of the G and B planes (after removing flash areas larger than 350 pixels). This mask is ANDed with the Salient point feature mask [13] of the image, which is finally ORed with the flash mask, obtained using the calculated threshold. It is followed by the verification and flash expansion steps explained in the Fixed Number Method (Section 2.2.4.1) to get the final flash mask.

Examples of flash masks for the image in Figure 2.2(b) for the standard deviation method with and without the salient feature mask are shown in Figure 2.7(a) and (b), respectively.



(a) (b)
Figure 2.7 Examples of flash mask for Standard Deviation Method
(a) with normal threshold and (b) with loose threshold in combination with the salient feature mask

2.2.4.3. Threshold calculation using curve-fitting equation. In this method, training and test sets of images are used. The desired thresholds are the thresholds found for manually marked flash regions in the training set of images. For finding the desired thresholds, histograms were plotted for each image for each color plane and were observed to manually to find the point where the histogram starts rising to reach the peak which represents the bright area in image. This value is taken as the desired threshold. Calculated thresholds are the thresholds found by the Otsu method. Using the calculated thresholds, polynomial curve fitting was performed to approximate them to the desired thresholds. It was found that ninth-order polynomial curves provided a better fit than lower-order polynomial curves, yielding 85% curve-fit accuracy. Using these equations the threshold is calculated for the test set images.

The equations for the curve fitting are as follows:

$$NewG_{cal} = (-7.368*10^{4})\cdot G_{cal}^{9} + (-3.551*10^{5})\cdot G_{cal}^{8} + (-7.323*10^{5})\cdot G_{cal}^{7} + (8.4*10^{6})\cdot G_{cal}^{6} + (-5.821*10^{5})\cdot G_{cal}^{5} + (2.459*10^{5})\cdot G_{cal}^{4} + (-5.956*10^{4})\cdot G_{cal}^{3} + (6488)\cdot G_{cal}^{2} + (123)\cdot G_{cal} + (-63.61) \quad (1)$$

$$NewB_{cal} = (-1.914*10^5)\cdot B_{cal}^9 + (8.963*10^5)\cdot B_{cal}^8 + (-1.834*10^6)\cdot B_{cal}^7 + (2.152*10^6)\cdot B_{cal}^6 + (-1.596*10^5)\cdot B_{cal}^5 + (7.762*10^5)\cdot B_{cal}^4 + (-2.476*10^5)\cdot B_{cal}^3 + (4.998*10^4)\cdot B_{cal}^2 + (-5797)\cdot B_{cal} + (294.7) \quad (2)$$

Using these G and B thresholds, the binary G and B flash masks are found and an OR operation is performed to get the final flash mask after removing flash areas larger than 350 pixels. It is followed by the verification and flash expansion steps as explained in the Fixed Number Method in Section 2.2.4.1 to get the final flash mask. After finding the flash mask by one of these methods explained in Sections 2.2.4.1, 2.2.4.2 and 2.2.4.3, the flash region is filled with a color-smoothing interpolation technique [14].

2.2.5. Color-smoothing Algorithm (Region Filling). After locating the flash regions, the next step is to color correct these regions. A Region Filling Algorithm is used from the previous work done by Shoba Umamaheshwaran [14]. In this technique, the smoothing direction is determined separately for each connected flash region. Based on the direction of the connected region for each pixel calculated, the appropriate flash pixel is replaced with these color values. This step is performed for every pixel in the boundary mask after which a new boundary mask is generated. This is performed iteratively until the entire flash region has been filled from the outside to the center of the region. The updating scheme is thus dependent on the proximity of a pixel to the surrounding non-flash regions.

3. GRAPHICAL USER INTERFACE (GUI)

For convenience of the users, a graphical user interface (GUI) application is designed using MATLAB® with the front end as shown in Figure 3.1.

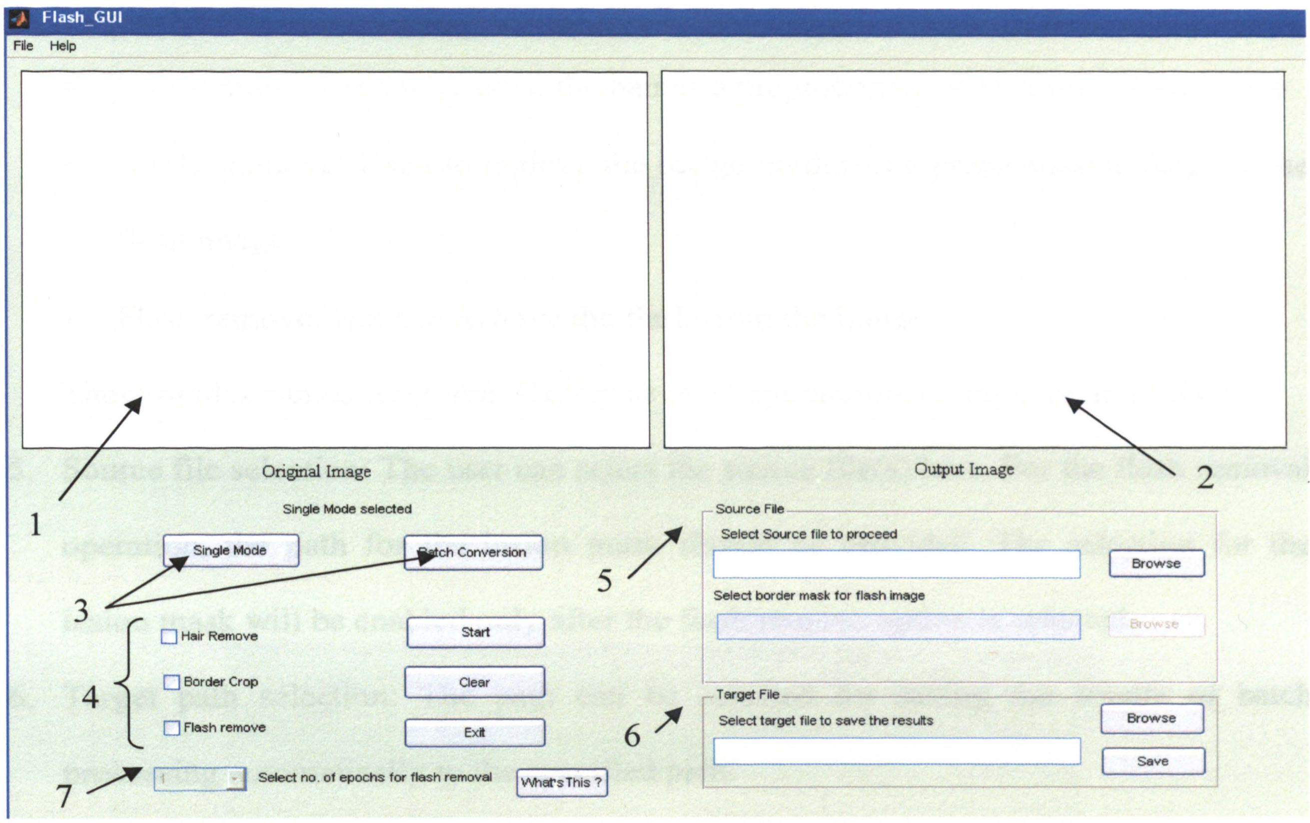


Figure 3.1 Layout of the GUI for flash removal

An overview of the items provided in the GUI is given as follows:

1. Frame to display the original image (for single mode)
2. Frame to display the output for processed image (for single mode)
3. Mode selection

- Single mode: only one image processed and result is displayed in the display frame.
- Batch mode: Batch of images is selected at a time and the results are stored at the specified path

4. Process modes:

- Hair remove: Used to remove the hair as a preprocessing step for the flash image
- Border remove: Used to remove the image border as a preprocessing step for the flash image
- Flash remove: Used to remove the flash from the image

These modes can be used individually or as a combination of any 2 or all of them.

5. Source file selection: The user can select the source file(s) here. For the flash removal operation, the path for the lesion mask should be provided. The selection for the lesion mask will be enabled only after the flash remove option is selected.
6. Target path selection: The path can be selected for saving the results of batch processing automatically to the specified path.
7. Epoch selection: the user can choose to process the image more than one time (1 to 5 times) in order to make sure that the flash is removed completely. However, multiple iterations take more time for processing.

This GUI allows the use of flash removal as well as the preprocessing steps required for it. Since different images have different characteristics, the user has to decide which steps to follow to get the best possible results. One example with all 3 steps involved for the single mode is shown in Figures 3.2, 3.3, 3.4, and 3.5. Figure 3.2 shows selecting single mode and checking all the check boxes.

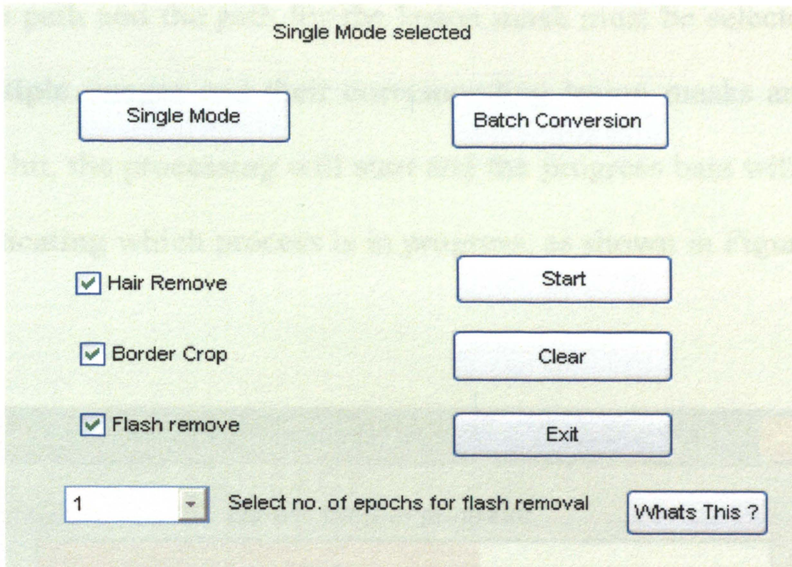


Figure 3.2 Mode and process selection for GUI

Figure 3.3 shows choosing the image(s) to be processed

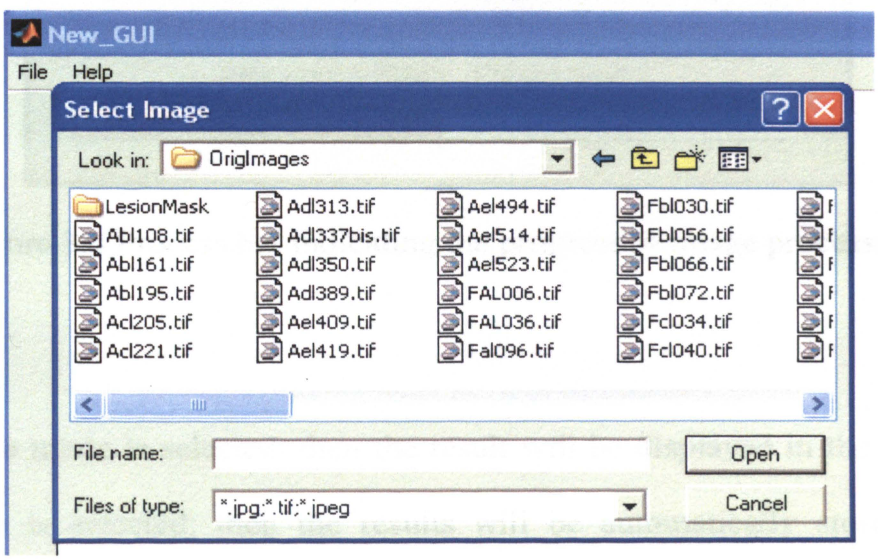


Figure 3.3 Image selection for the GUI

The image path and the path for the lesion mask must be selected. In the case of Batch mode, multiple images and their corresponding lesion masks are selected. After the start button is hit, the processing will start and the progress bars will be displayed one after the other indicating which process is in progress, as shown in Figure 3.4

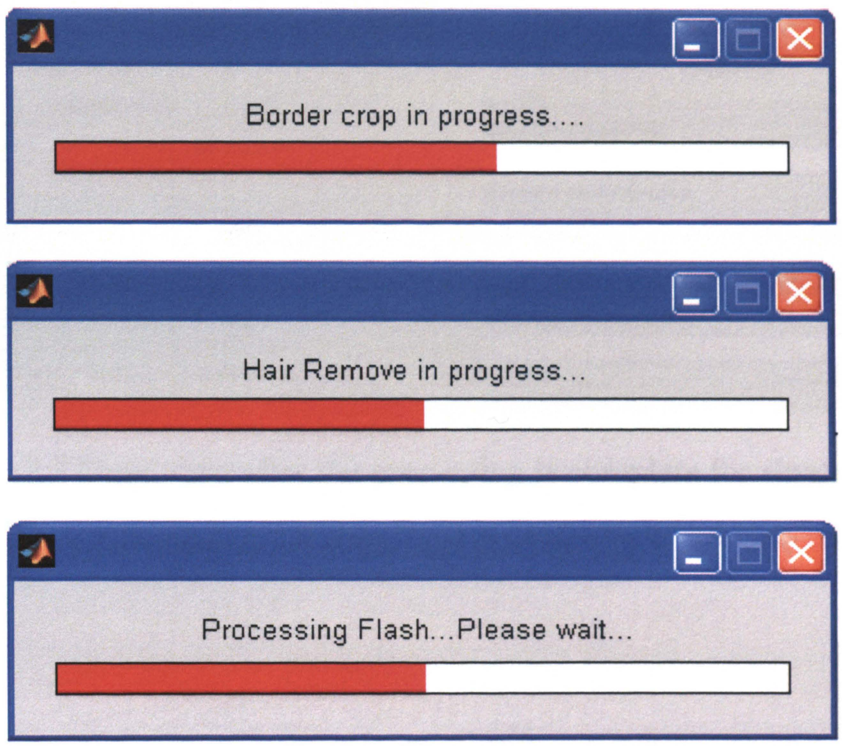


Figure 3.4 Process bar indicating the progress of image processing

If single mode is selected, then the result will be displayed in the display frame. If batch mode is selected, then the results will be automatically stored at the path specified as the target path. The results with an example of single mode processing is shown in Figure 3.5.

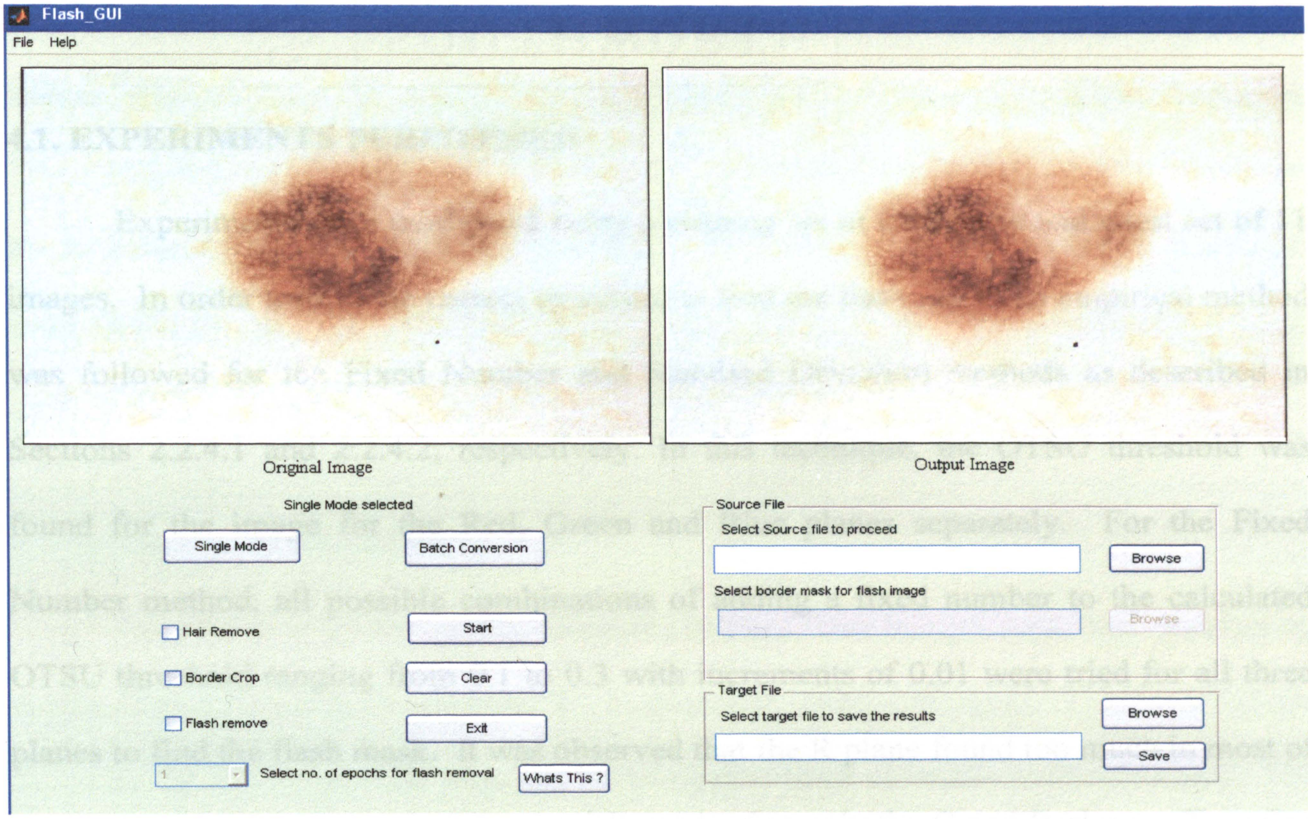


Figure 3.5 Final view after the processing is complete for single mode

4. RESULTS

4.1. EXPERIMENTS PERFORMED

Experiments were performed using a training set of 85 images and a test set of 11 images. In order to find the correct threshold to find the flash mask, an empirical method was followed for the Fixed Number and Standard Deviation methods as described in Sections 2.2.4.1 and 2.2.4.2, respectively. In this technique, the OTSU threshold was found for the image for the Red, Green and Blue planes separately. For the Fixed Number method, all possible combinations of adding a fixed number to the calculated OTSU threshold ranging from 0.1 to 0.3 with increments of 0.01 were tried for all three planes to find the flash mask. It was observed that the R plane found too much in most of the cases. So the same procedure was followed using only the G and B planes. To refine the results, the properties of the lesion area like the mean of the lesion area and the standard deviation were observed. It was observed that for more pinkish images the threshold condition should be different than for other images (discussed later in Section 4.1). The average true positive rate was calculated using the pixel-based and region-based methods (described in the next section) for each output set for the above combinations and the one which gave the best results was finalized as the fixed threshold for this method.

A similar procedure was followed for the standard deviation method where instead of adding a fixed number; various combinations of fractions (multiples) of the standard deviations (ranging between 0.1 to 1.4 times the standard deviation with an increment of 0.1) were added to the calculated Otsu thresholds.

For the curve fitting method, the MATLB® curve-fit tool was used. It offers various types of curve fit methods including polynomial of different degrees, Gaussian, Smoothing spline, etc. This tool was used to choose the appropriate equation which fits the calculated Otsu thresholds and the desired thresholds, where the desired thresholds were found by observing the histograms for the lesion areas for the manually marked flash images. The equations are given in Section 4.3.

4.2. SCORING OF RESULTS

In order to score the results, manually marked flash images were used (marked by Austin Bangert and checked by Dr. W. V. Stoecker, a dermatologist). The flash regions found by the above methods were compared against these manually marked images. The scoring of the results was done on the test set of 11 images. The following two methods were used for scoring :

i. Pixel-based method
ii. Region-based method

4.2.1. Pixel-based Method. Here, the true positive rate is calculated as:

$$\%TP = \frac{\begin{array}{c}\text{No. of flash pixels detected by the flash program which are also}\\ \text{present in the manually marked flash}\end{array}}{\text{No. of flash pixels in manually marked flash}} \times 100$$

The false positive rate is calculated as

$$\%FP = \frac{\begin{array}{c}\text{No. of flash pixels detected by the flash program which are not}\\ \text{present in the manually marked flash}\end{array}}{\text{No. of flash pixels comprising the manually marked flash}} \times 100$$

4.2.2. Region-based Method. Here the true positive rate is calculated as:

$$\%TP = \frac{\text{No. of flash areas detected by the flash program which are present in manually marked flash}}{\text{No. of flash areas comprising the manually marked flash areas}} \times 100$$

where a flash area is said to be detected if at least 10% of its pixels are found by the flash program. For example, if in a manually marked flash mask 100 flash regions are present and out of them 80 are detected then the percentage true positive is 80%.

The number of false positive detections is the number of flash regions found by the flash detection algorithm which are actually not present in the manually marked flash mask.

Both scoring methods were applied to the lesion area and the whole image separately to determine the percentages of accuracy. A summary of the results of scoring the training set of 85 images is given in Table 4.1. The results for each image in training set are summarized in Appendix F.

Table 4.1 Summary table for results of scoring for the 85 images in the training set

Method	Pixel based scoring				Region Based scoring			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Fixed number	41.75	2.30	31.15	0.35	51.41	360.66	42.32	122.13
Fixed number and SP	52.42	2.80	41.29	0.43	64.72	440.39	52.96	145.49
Std Dev	52.22	2.77	41.15	0.42	64.64	422.13	52.87	131.11
Std Dev and SP	54.78	2.95	54.88	1.28	69.72	707.34	68.62	453.52
Curve fit	64.06	3.93	65.43	0.99	83.86	872.08	85.41	324.55

It was observed that the curve fitting method gave best results for the training set but it did not work that well on a large test set. (These results will be presented later.)

This was probably because of the over fitting during the training. Also it worked really well on a few images, however it completely failed on some of them. This is mainly because of the inconsistent nature of the fitting curve as shown in Figure 4.1.

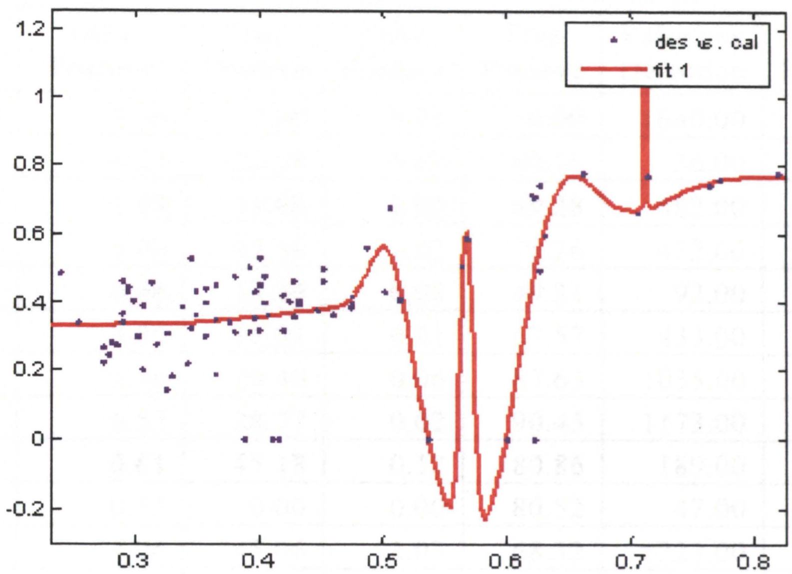


Figure 4.1 Curve fitting equation for green plane of test set images using MATLAB® curve fitting tool

As seen in Figure 4.1, there are sudden drops and spikes in the curve which explain the complete failure of the Curve Fitting Method on a few images due to incorrect threshold calculation. It might be possible to achieve better accuracy using a larger training set which can give a more accurate curve fitting equation.

The Fixed Number Method generated acceptable results but the Standard Deviation Method gave consistently better results on the training as well as the test set. The results are improved when it is combined with the Salient Point method.

The summary of results on a small training set of 11 images is given in Tables 4.2, 4.3, 4.4, 4.5, and 4.6.

Table 4.2 Summary of results for Fixed Number Method

Image	Score Method (Pixels)				Score Method (Area)			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Adl395	0.00	13.06	0.00	0.01	0.00	1640.00	0.00	13.00
FAL044	49.33	0.23	22.28	0.00	60.16	36.00	35.19	2.00
FBL026	44.10	1.59	14.60	0.00	69.28	482.00	22.58	1.00
Fal100	54.03	5.00	43.58	0.02	79.26	472.00	62.50	12.00
Fbl028	29.37	0.46	15.24	0.05	49.31	92.00	35.86	21.00
Fcl026	46.31	3.65	13.01	0.01	67.57	433.00	28.57	2.00
Fcl100	52.42	5.46	29.40	0.06	87.63	1035.00	50.00	35.00
Fll069	48.87	6.53	28.77	0.02	90.45	1173.00	65.85	16.00
Fll128	40.34	0.61	45.18	0.12	80.86	189.00	77.48	43.00
Nhl013	66.03	0.53	0.00	0.00	80.52	47.00	0.00	0.00
Nml004	37.69	4.06	45.06	3.03	68.32	1223.00	75.64	1042.00
Average	46.85	2.81	25.71	0.33	73.33	518.20	45.37	117.40

Table 4.3 Summary of results for Fixed Number Method combined with Salient feature masks

Image	Pixel Based Scoring				Area Based Scoring			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Adl395	0.00	13.06	0.00	0.008	0.00	1640	0.00	13
FAL044	51.05	0.25	22.82	0.002	60.55	37	35.19	2
FBL026	44.62	1.59	14.78	0.004	69.28	486	22.58	2
Fal100	54.49	4.96	43.77	0.017	80.00	475	62.50	12
Fbl028	29.67	0.45	15.49	0.046	49.54	94	36.36	21
Fcl026	46.71	3.66	14.04	0.008	67.57	445	28.57	2
Fcl100	53.74	5.44	31.57	0.063	87.63	1038	50.00	35
Fll069	50.11	6.46	29.87	0.015	90.91	1171	68.29	16
Fll128	40.48	0.59	45.39	0.122	81.14	192	78.38	44
Nhl013	66.07	0.54	0.00	0.000	80.52	51	0.00	0
Nml004	37.71	4.06	45.09	3.030	68.32	1226	75.64	1045
Average	47.46	2.80	26.28	0.33	73.54	521.50	45.75	117.90

Table 4.4 Summary of results for Standard Deviation Method

Image	Pixel Based Scoring				Area Based Scoring			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Adl395	0.00	12.73	0.00	0.01	0	1468	0	4
FAL044	35.86	0.80	62.68	0.71	37.5	388	72.2222	387
FBL026	8.64	0.17	34.85	0.16	18.072	73	58.0645	72
Fal100	45.03	1.08	49.22	0.12	71.852	224	81.25	44
Fbl028	29.78	1.54	32.38	1.37	52.546	581	60.6061	558
Fcl026	35.33	1.38	21.92	0.22	55.405	302	47.619	129
Fcl100	47.82	2.34	53.98	0.80	72.165	487	55.5556	327
Fll069	38.91	2.89	39.50	0.20	65	348	56.0976	66
Fll128	30.03	0.57	36.23	0.37	62	168	72.973	120
Nhl013	60.31	0.79	0.00	0.48	79.221	231	0	216
Nml004	37.49	4.60	44.80	3.60	70.297	1462	78.2051	1226
Average	36.92	1.62	37.56	0.80	58.41	426.40	58.26	314.50

Table 4.5 Summary of results for Standard Deviation Method combined with Salient feature masks

Image	Pixel Based Scoring				Area Based Scoring			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Adl395	0.00	12.74	0.00	0.01	0.00	1470.00	0.00	6.00
FAL044	36.01	0.86	64.16	0.76	37.50	389.00	72.22	388.00
FBL026	8.69	0.18	35.22	0.17	18.07	74.00	58.06	73.00
Fal100	45.17	1.11	50.58	0.14	71.85	225.00	81.25	45.00
Fbl028	30.80	1.71	35.12	1.49	52.55	572.00	60.61	549.00
Fcl026	35.40	1.39	22.26	0.23	55.41	305.00	47.62	132.00
Fcl100	48.36	2.40	57.83	0.85	72.16	505.00	55.56	345.00
Fll069	39.03	2.95	40.37	0.22	65.00	346.00	56.10	64.00
Fll128	30.03	0.63	36.25	0.39	62.00	175.00	72.97	126.00
Nhl013	60.31	0.81	0.00	0.49	79.22	236.00	0.00	221.00
Nml004	38.16	4.63	45.68	3.61	70.30	1461.00	78.21	1224.00
Average	37.20	1.67	38.75	0.83	58.41	428.80	58.26	316.70

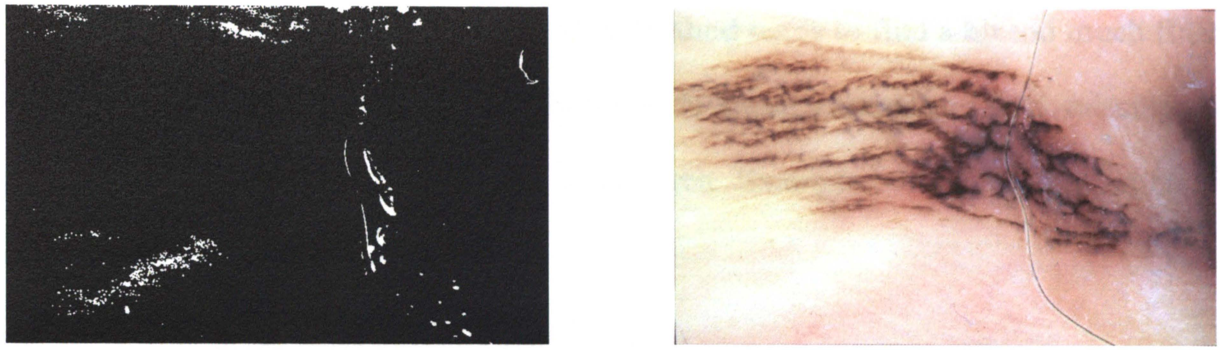
Table 4.6 Summary of results for Curve Fitting Method

Image	Pixel Based Scoring				Area Based Scoring			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Adl395	0.00	12.79	0.00	0.02	0.00	1521.00	0.00	15.00
FAL044	45.77	2.89	73.15	0.47	74.61	1216.00	100.00	271.00
FBL026	40.22	2.99	34.31	0.05	66.87	1505.00	64.52	55.00
Fal100	53.42	4.33	52.92	0.09	93.33	1153.00	93.75	41.00
Fbl028	44.55	2.59	49.74	1.02	88.89	1278.00	91.92	538.00
Fcl026	48.70	2.91	37.67	0.10	82.43	884.00	66.67	93.00
Fcl100	50.52	2.23	60.96	0.39	86.60	463.00	88.89	130.00
Fll069	52.57	5.77	49.23	0.15	92.27	1147.00	80.49	64.00
Fll128	41.94	0.99	49.20	0.32	86.00	392.00	90.09	157.00
Nhl013	64.39	3.47	50.00	0.13	94.81	1012.00	100.00	106.00
Nml004	49.57	5.59	54.63	3.95	84.16	1245.00	87.18	907.00
Average	49.17	3.38	51.18	0.67	85.00	1029.50	86.35	236.20

Note: The image Adl395 has no flash region, so the denominator for calculating the true positive rate (for both pixel-based and region-based method) is zero, which gives the true positive rate to be actually an infinite number. Hence, that value is shown as '0' in the table to indicate that it is not applicable to count the true positive rate in such case.

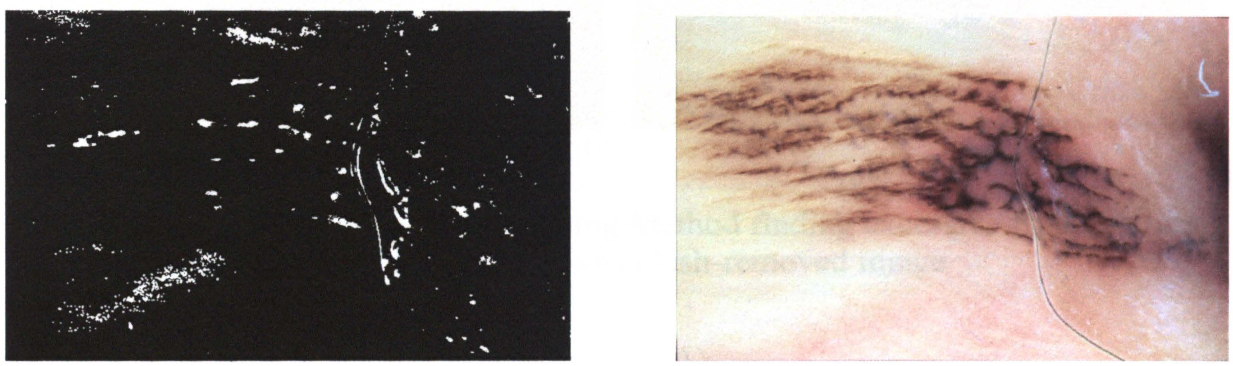
Figure 4.2 shows an example image from the training set. Figures 4.3, 4.4 and 4.5 show the flash masks found by each method for this image.

Figure 4.2 Example Image (Abl108.tif) from training set



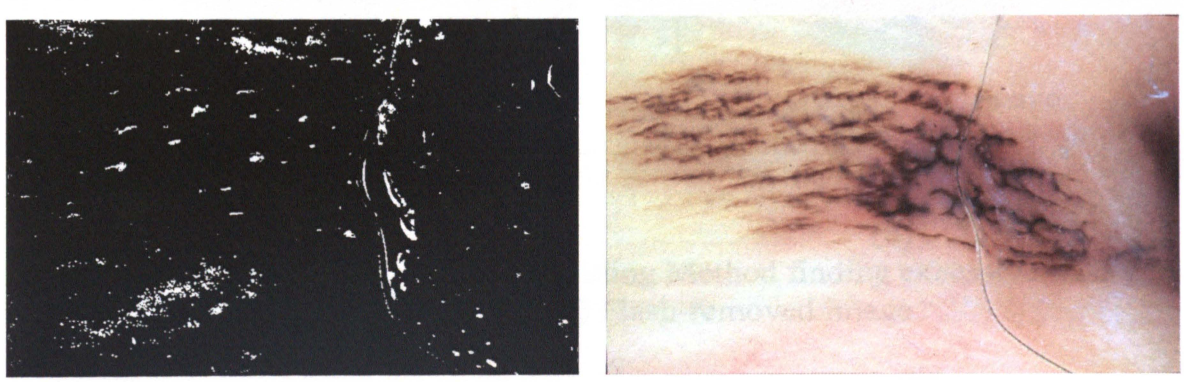
(a) (b)

Figure 4.3 Example of flash removal for the Fixed Number Method (a) The flash mask and (b) The flash-removed image.



(a) (b)

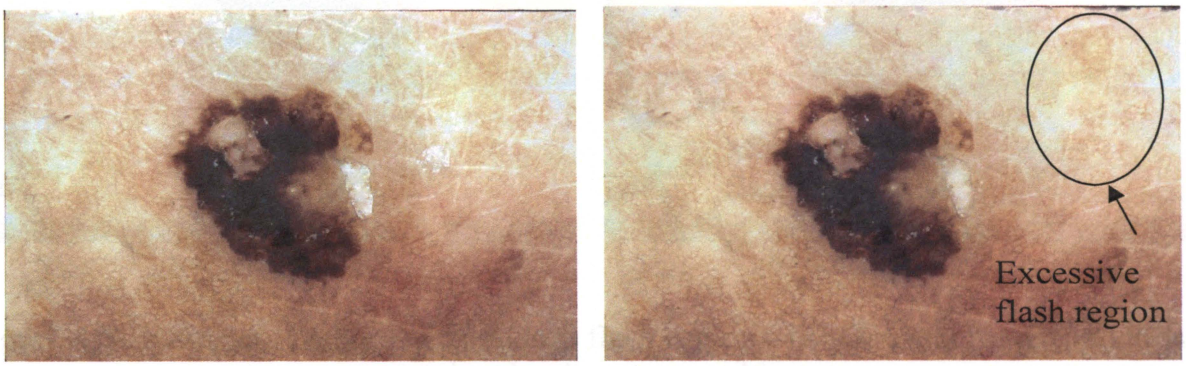
Figure 4.4 Example of flash removal for Standard Deviation Method (a) The flash mask and (b) The flash-removed image.



(a) (b)

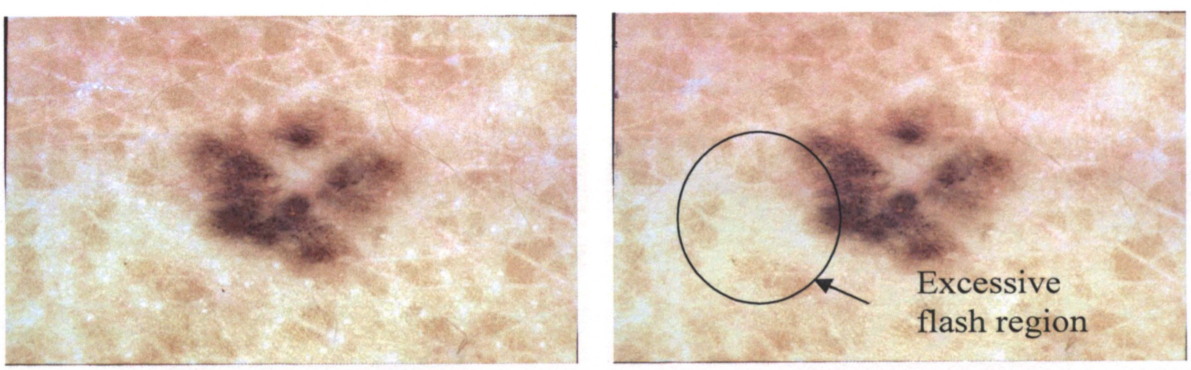
Figure 4.5 Example of flash removal for Curve Fitting Method (a) The flash mask and (b) The flash-removed image.

It was observed that the Curve Fitting Method tends to find a bit too much flash as compared to the other two methods, but this is more acceptable than missing flash regions as long as it does not affect the lesion area. Examples of finding too much flash are shown in Figures 4.6 and 4.7.



(a) (b)

Figure 4.6 Example of the Curve Fitting Method finding excessive flash regions. (a) Original image. (b) Flash-removed image

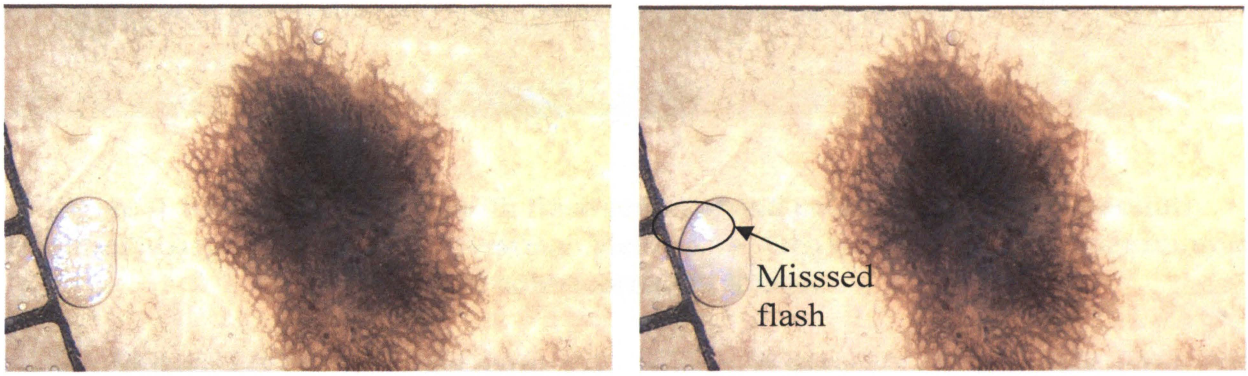


(a) (b)

Figure 4.7 Another example of the Curve-Fitting Method finding excessive flash regions. (a) Original image (b) Flash-removed image.

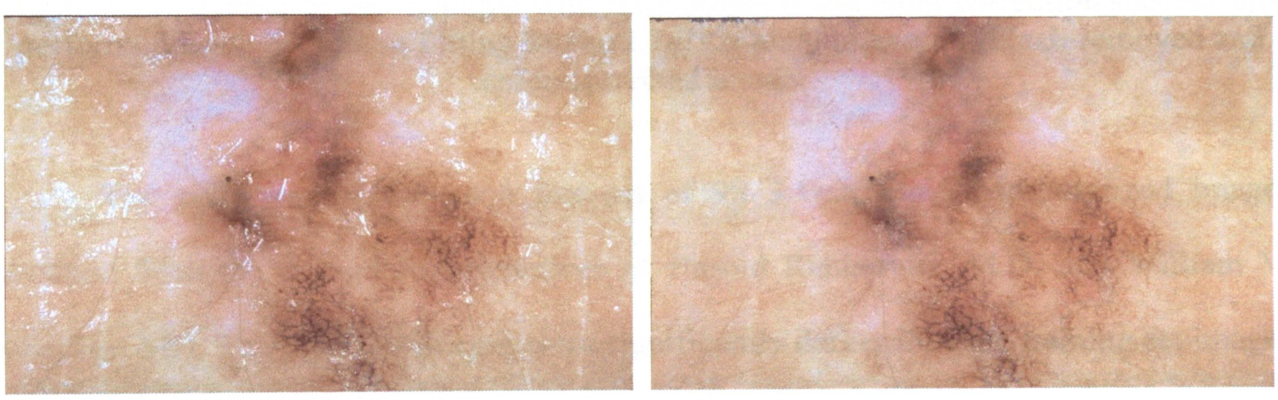
As seen in the above example, sometimes the lighter skin area around the lesion was found as a flash area and was filled, but this did not distort the lesion area.

In some cases some obvious flash areas are missed as the value of the thresholds are not perfect (curve-fit equation is 85% accurate). An example is shown in Figure 4.8.

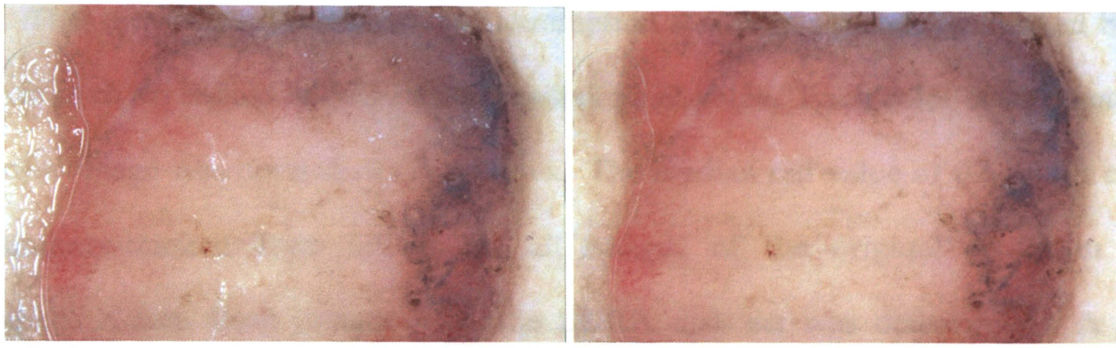


(a) (b)
Figure 4.8 Example of image with missed flash regions. (a) Original image. (b) Flash-removed and color-corrected image.

A few more examples of flash-removed images with the Standard Deviation and Curve Fitting Method are shown in Figures 4.9 to 4.11.

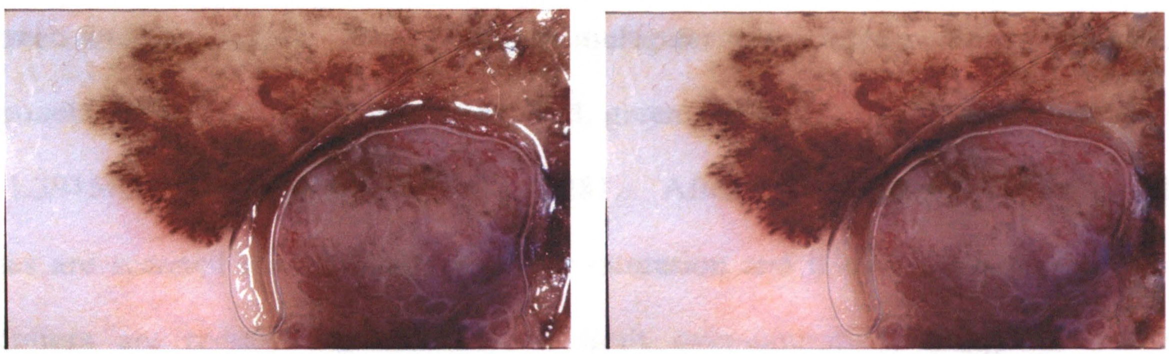


(a) (b)
Figure 4.9 Image example with flash removal and color correction successfully performed using the Curve Fitting Method. (a) Original image. (b) Flash-removed and color-corrected image.



(a) (b)

Figure 4.10 Image example with flash removal and color correction successfully performed using the Standard Deviation Method. (a) Original image. (b) Flash-removed and color-corrected image.



(a) (b)

Figure 4.11 Another image example with flash removal and color correction successfully performed using the Curve Fitting Method. (a) Original image. (b) Flash-removed and color-corrected image.

The code was tested on a test set of 99 3-Gen Fluid non-polarized images obtained from the dermatology practices of Harold Rabinovitz, M.D. and William V. Stoecker, MD. The received images were significantly darker than the training set images which were obtained using a Heine Dermaphot camera attachment, so the flash removal code did not work well on them. These 3-Gen images were modified using a normalization technique developed by Mark Wronkiewicz and Raeed Chowdhury.

This normalization rests on the assumption that pixels manually marked by flash masks in each set should have a similar median pixel value. Each color component of the new color image was normalized as follows. The median red value of the pixels covered by manually-marked flash masks for all images of the original 85-lesion set of Dermaphot Xenon flash images from the EDRA disk set was determined. A similar technique was performed for the new 3-Gen Fluid non-polarized set of images obtained in the clinics as above. The median red value obtained from the EDRA set was divided by the median red pixel value of the 3-Gen Fluid image pixels covered by flash masks to give a red multiplier for all red pixels in the image. A similar operation was performed for green and blue pixels. Thus the image multiplier was a three-component vector, one component for each of the three colors: red, green and blue. The three multipliers were R = 1.39156, G = 1.46642, and B = 1.33281. After these multipliers are applied, the images are scanned automatically for over-saturation and if over-saturation occurs, the multipliers are reduced dynamically to limit over-saturation, using the multiplier reduction algorithm of Chowdhury-Wronkiewicz in Appendix G. An acceptable saturation limit was empirically optimized, based on X, the total number of red, green and blue pixels that had intensity = 255. After inspecting 200 images, X was optimized to 390,000 values for a 1024 X 768 image. This is the limit now used in the automatic saturation scanning.

A summary of the results of scoring the test set of 99 images is given in Table 4.7. The results for each image in test set are summarized in Appendix H.

Table 4.7 Summary table for results of scoring for the 99 images in the test set

Method	Pixel-based scoring				Region-Based scoring			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Fixed number	22.53	1.30	24.27	0.82	40.37	965.33	42.14	620.48
Fixed number and SP	23.04	1.60	25.06	0.90	41.22	1014.60	42.21	583.88
Std Dev	25.98	2.27	25.88	1.41	46.84	1550.55	47.23	922.78
Std Dev and SP	26.25	2.27	26.17	1.42	47.89	1591.29	48.44	948.82
Curve fit	20.87	1.11	24.30	0.49	38.56	668.48	40.52	275.56

Note : The False positive detection values for region-based scoring in Tables 4.1 and 4.2 are fractions because these are average values for the entire set. False positive detection values are the count of the falsely detected areas and hence are integer values as given in all the tables in Appendix F and G. Some cells in the tables in Appendix F and G are blank because the values calculated for them were NaN (short form for Not a Number in XLS) due to zero denominators in the formula for calculating TP values. The denominator is zero when there is no flash area in the image so that the pixel count for manually marked flash is zero. Also there are few zeros in the FP column in the tables in Appendix F and G. These are because of a numerator equal to zero which happens when all the flash pixels found by the flash removal code are also present in the manual flash marks.

5. CONCLUSION

It was observed that the Standard Deviation Method performed better than the other two methods. The Fixed Number Method gives intermediate accuracy and consistency in results. The Curve Fitting Method works very well on a few images, especially when they are similar to the images in training set, however it completely fails on a few images. This is because the curve fitting is not 100% accurate on the given points. Results of curve fitting can perhaps be improved using a larger training set. Also it is observed that sometimes the Standard Deviation method finds a little too much area as flash, especially when the skin surrounding the lesion is very light, but it is acceptable as long as it does not distort the lesion.

This flash removal can be used as a preprocessing step to improve the visibility of the images so that the feature extraction will be more accurate making the diagnosis more precise.

APPENDIX A

MAIN PROGRAM FOR FLASH FINDING
FinalFlash.m

```
%****************************************************************/
%* Main program for flash finding
%* Program       : FinalFlash.m
%* Author                    : Radhika Nagane
%* Input        : Original Image and lesion mask for it
%* Output                    : Flash removed image(FinalImage)
%* Description   : This program is main program for flash detection
%* Which calls the subroutines for the intermediate steps involved in
%* flash removal
%****************************************************************/
I1 = imread('C:\OrigImages\img.tif');  %path for original image
I2 = imread('C:\BorderMask\img_mask.tif'); %path for lesion mask
[m n] = size(I2);
%converting image to relative color and finding threshold with graythresh
I = I1; %saving the original color image to display later
%generating the relative color image
eval(['cd ' 'C:\Flash\FinalCode']); %path where code files are located
I1 = RelColCalculate(I1,I2);
%call Flash Mask finding routine
%flashRegion contains the original flash region, flashRgnLuse
%contains the flash area found with much loose threshold
[flashRegion flashRgnLuse] = FindFlash(I1,I,I2);
%get the mask generated from the salient feature code and combine with
%the flash mask
fn = ['C:\Flash\SalientMask\' f1(1:end-4),'.tif'];%path for salient
%feature masks
SF_mask = imread(fn);
%combining the loose OTSU flash mask with the salient feature mask and then
%combining the resultant mask with my original mask for method 2
Flash = flashRegion | (SF_mask and flashRgnLuse);
%----call the region filling routine to fill the flash regions
FinalImage = flashremoval(I1,I,I2,Flash);
```

APPENDIX B

CALCULATING THE RELATIVE COLOR IMAGE
RelColCalculate.m

```
%***************************************************************/
%* Subroutine to find the reative colour image
%* Program                 : RelColCalculate.m
%* Author                  : Kapil Gupta
%* Input                   : Original Image and lesion mask for it
%* Output                  : Relative color image
%***************************************************************/
function[colImg] = RelColCalculate(OrigImg,Mask)
colImg = double(OrigImg);
colImg1 = OrigImg;
[m n w]=size(img);
[L NUM] = bwlabel(img);
Temp = regionprops(L,'Area');
area = Temp.Area;
backGround = size(img,1)*size(img,2)-area;
dist = bwdist(img);
dist1 =0; dist2=0;
lowerArea =0.1*area;
upperArea = 0.2*area;
cumArea =0;
%DETECT THE INNER BOUNDARY
while ((cumArea < lowerArea) andand (cumArea < backGround))
   dist1=dist1+1;
   cumArea =0;
   for tmp_y = 1:size(img,1)
      for tmp_x =1:size(img,2)
         if((dist(tmp_y,tmp_x) <= dist1) andand (dist(tmp_y,tmp_x) > 0))
            cumArea = cumArea+1;
         end
      end
   end
end
cumArea = 0;
```

```
%DETECT THE OUTER BOUNDARY
while ((cumArea < upperArea) andand (cumArea < backGround))
   dist2=dist2+1;
   cumArea =0;
   for tmp_y = 1:size(img,1)
      for tmp_x =1:size(img,2)
         if((dist(tmp_y,tmp_x) <= dist2) andand (dist(tmp_y,tmp_x) > 0))
            cumArea = cumArea+1;
         end
      end
   end
end
outImg = ones(size(img))*255;
if(backGround < lowerArea)
   outImg = img;
else
   for tmp_y =1:size(img,1)
      for tmp_x =1:size(img,2)
         if((dist(tmp_y,tmp_x)>=dist1)andand(dist(tmp_y,tmp_x) <= dist2))
            outImg(tmp_y,tmp_x)=0;
         end
      end
   end
end
[rows cols] = find(outImg ==0);
red =0; green=0; blue =0;
for tmp_x = 1:length(rows)
   a = colImg1(rows(tmp_x),cols(tmp_x),1);
   b = colImg1(rows(tmp_x),cols(tmp_x),2);
   c = colImg1(rows(tmp_x),cols(tmp_x),3);
   % The if condition is added for non skin removal
   if a < 250 and b < 250  and c < 250 and a > b and a > c and a > 100
      red = red + double(colImg(rows(tmp_x),cols(tmp_x),1));
```

```
        blue = blue + double(colImg(rows(tmp_x),cols(tmp_y),2));
        green = green + double(colImg(rows(tmp_x),cols(tmp_y),3));
    end
end
R = round(red/length(rows));
G = round(green/length(rows));
B = round(blue/length(rows));
rel_col_img = zeros(size(colImg)); %stores the relative color image
rel_col_img(:,:,1)=colImg(:,:,1)-R;    %separating the R G B planes
rel_col_img(:,:,2)=colImg(:,:,2)-G;
rel_col_img(:,:,3)=colImg(:,:,3)-B;
colImg = rel_col_img;
```

APPENDIX C

SUB-ROUTINE FOR FINDING FLASH MASK

FindFlash.m

```
**************************************************************/
%* Subroutine for flash mask finding
%* Program       : FindFlash.m.m
%* Author                    : Radhika Nagane
%* Input       : Original Image, Relative color image and lesion mask
%* Output                    : Flash mask
%* Description    : This subroutine finds the flash mask for fix number method. It is different for
%the different methods like fix number methos, standard deviation method
%and the curve fit method.
%**************************************************************/
function [flashRegion flashRgnLuse] = FindFlash(rel_col_img,colImage,BorderMask)
I1 = rel_col_img;
I2 = BorderMask;
I = colImage;
[m n] = size(I2);
Red = (I1(:,:,1)+255)/512;
Green = (I1(:,:,2)+255)/512;
Blue = (I1(:,:,3)+255)/512;
%---normalizing complete. I2 is relative color normalised image, finding the flash region using OTSU
[r c] = find(I2 ~= 0);
histR = zeros(1,length(r));
histG = zeros(1,length(r));
histB = zeros(1,length(r));
cnt = 1;
for mm = 1 : m
for nn = 1:n
    if I2(mm,nn)>0
        histR(:,cnt) = Red(mm,nn);
        histG(:,cnt) = Green(mm,nn);
        histB(:,cnt) = Blue(mm,nn);
        cnt = cnt + 1;
    end
end
```

```
end
levelR1=graythresh(histR);                %obtaining the threshold level based on blue color plane
levelG1=graythresh(histG);
levelB1=graythresh(histB);
clear histR histG histB
% % Finding threshold by fix nnumber method
levelG = 0;
levelB = 0;
if levelB1 < 0.5
levelG = levelG1 + 0.25;
levelB = levelB1 + 0.25;
elseif levelG1 > 0.5 and levelB1 > 0.5
levelG = levelG1;
levelB = levelB1;
elseif levelR1 > 0.85 and levelG1 > 0.5 and levelB1 > 0.5
levelG = levelG1 + 0.1;
levelB = levelB1 + 0.1;
end
th_g=im2bw(Green,max(levelG,0));%Thresholding the Image by taking a level equal to levelG
th_b=im2bw(Blue,max(levelB,0)); %Thresholding the Image by taking a level equal to levelB
%---applying the criterion for large flash region to eliminate very
%large areas which are found as flash
th_g = RemLargeFlash(th_g);
th_b = RemLargeFlash(th_b);
flashRegion = th_b | th_g;
[r c] = find(flashRegion > 0);  % conidering entire region
%added to find the flash region in the images in which it didnt find much.
%when it doesnt find much, only blue plane is considered
if length(r) < 50
flashRegion = th_b;
end
% %---again applying the criterion for large flash region to eliminate very
% %large areas which are found as flash
```

```
flashRegion = RemLargeFlash(flashRegion);
% condition added to find pure white region.....this is done because some
% complete white regions are getting removed because in the test images
% there were some large white flashes and they were getting rmoved because of
% the constraint on the flash area
[r c] = find(I(:,:,1) > 245 and I(:,:,2) > 240 and I(:,:,3) > 225);
flashRegion1 = zeros(m,n);
for j = 1 : length(r)
flashRegion1(r(j),c(j)) = 255;
end
% applying the criterion for large flash region to eliminate very
% large areas which are found as flash while finding pure white region
flashRegion1 = RemLargeFlash(flashRegion1);
flashRegion = flashRegion1 | flashRegion;
clear flashRegion1
%---using 40*40 window around flash pixels found in findFlash to compare
%pixel value with with averageh of the window
[xcount,ycount] = find(flashRegion>0);
for k = 1:length(xcount)
if xcount(k)>20 and xcount(k)<m-20 and ycount(k)>20 and
ycount(k)<n-20      %to avoid 40 pixel area near image border
temp = I(xcount(k)-20:xcount(k)+20,ycount(k)-
20:ycount(k)+20,1:3);
If I(xcount(k),ycount(k),1)<mean(mean(temp(:,:,1)))|I(xcount(k),ycount(k),2)< ean(mean(temp(:,:,2)))
| I(xcount(k),ycount(k),3) < mean(mean(temp(:,:,3)))
%if I1(xcount(k),ycount(k),3) < mean(mean(temp(:,:,3)))
flashRegion(xcount(k),ycount(k),1) = 0;
end
end
end
clear k
%finding the pixels near the flash region using the methos sugegsted by Dr Stoecker.
SE=strel('disk',1);
```

```
flashRegion1 = imdilate(flashRegion,SE);  %dilating the flash region to include more area in flash region
as suggested by Dr Stoecker
flashBorder = flashRegion1 and (~flashRegion);
clear flashRegion1
[r,c] = find(flashBorder > 0);
cnt = 0;
Fillvalue = max(max(flashRegion));    %used because sometimes white area is marked with 1 and
sometimes its 255
for k = 1:length(r)
if r(k)>20 and r(k)<m-20 and c(k)>20 and c(k)<n-20
if I(r(k),c(k),1) > 200 and I(r(k),c(k),2) > 200 and I(r(k),c(k),3) > 200
flashRegion(r(k),c(k)) = Fillvalue;
cnt = cnt + 1;
end
end
end
clear Fillvalue flashBorder th_g th_b th_r levelG levelB
%finding the flash region with relaxed OTSU
flashRgnLuse = zeros(size(flashRegion));
levelG = 0;
levelB = 0;
if levelB1 < 0.5
levelG = levelG1 + 0.24;
levelB = levelB1 + 0.24;
elseif levelG1 > 0.5 and levelB1 > 0.5
levelG = levelG1-0.1;
levelB = levelB1-0.1;
elseif levelR1 > 0.85 and levelG1 > 0.5 and levelB1 > 0.5
levelG = levelG1 + 0.095;
levelB = levelB1 + 0.095;
end
th_g=im2bw(Green,max(levelG,0));  %Thresholding the Image by taking a level equal to levelG
th_b=im2bw(Blue,max(levelB,0));  %Thresholding the Image by taking a level equal to levelB
```

```
%---applying the criterion for large flash region to eliminate very
%large areas which are found as flash
th_g = RemLargeFlash(th_g);
th_b = RemLargeFlash(th_b);
flashRgnLuse = th_b | th_g; %method 1 : manual thresholds, use
only G and B plane for finding flash mask
```

APPENDIX D

SUB-ROUTINE TO REMOVE THE LARGE FLASH AREAS DETECTED MISTAKENLY

RemLargeFlash.m

```
%****************************************************************/
%* Subroutine to remov flash regions > 350 pixels
%* Program            : RemLargeFlash.m
%* Author             : Radhika Nagane
%* Input    : Flash mask
%* Output             : Flash mask with large areas removed
%* Description   : Sometimes very light skin color is detected as flash, to
%remove it thie sub routine is used
%****************************************************************/
function LargeAreaRem = RemLargeFlash(Image)
LargeAreaRem = Image;
[Im,num] = bwlabel(LargeAreaRem);
  area = regionprops(Im,'Area');
  for cnt = 1:num
    a = cell2mat(struct2cell(area(cnt)));
     if a > 350
      [mm,nn] = find(Im == cnt);
      for cc = 1:length(mm)
        LargeAreaRem(mm(cc),nn(cc)) = 0;
      end
    end
  end
  clear area Im num cnt a mm nn
```

APPENDIX E

SUB-ROUTINE FOR FILLING THE FLASH REGION
FillFlash.m

```
%*****************************************************************/
%* Subroutine to fill flash areas with surrounding color
%* Program                : flashremovel.m
%* Author                 : Shoba Umamaheshwaran
%* Input        : Flash mask and Original Image
%* Output                 : Flash removed final image
%* Description    : This code fills the flash areas found by the flash mask
%to give final color smoothed image
%*****************************************************************/function
[nonFlashImg] = flashremoval(rel_col_img,colorImage,BorderMask,flashRegion)
h = waitbar(0,'Processing Flash...Please wait...');
[d e f] = size(colorImage);
Image = double(255*ones(d+10,e+10,3));  %to add a border around the image to be able to consider
the flash pixels near the border also
colImage = double(255*ones(d+10,e+10,3));
lesion_area = double(zeros(d+10,e+10));
BW2 = double(zeros(d+10,e+10));
%adding border of 5 pixels on each side of image so that areas near the
%boredr are also included in the flash region
[a b c] = size(colImage);
Image(6:a-5,6:b-5,1:3) = rel_col_img;          %relative color image
colImage(6:a-5,6:b-5,1:3) = colorImage;          %original color iamge
lesion_area(6:a-5,6:b-5) = BorderMask;        %lesion mask
BW2(6:a-5,6:b-5) = flashRegion;               %Flash region
lesion_area = double(lesion_area);
Image = double(Image);
colImage = double(colImage);
BW2 = double(BW2);
[rows,cols,dim]=size(Image);
  %Gradient Calculation using Sobel Operator
sh=fspecial('sobel');                  %Horizontal structuring element
sv=sh';                                %Vertical Structuring  element
   %Filtering Each Color Plane in x and y direction
```

```
Rx=imfilter(Image(:,:,1),sh,'replicate');
Ry=imfilter(Image(:,:,1),sv,'replicate');
Gx=imfilter(Image(:,:,2),sh,'replicate');
Gy=imfilter(Image(:,:,2),sv,'replicate');
Bx=imfilter(Image(:,:,3),sh,'replicate');
By=imfilter(Image(:,:,3),sv,'replicate');
%Compute Parameters of the vector gradient
gxx=Rx.^2+Gx.^2+Bx.^2;
gyy=Ry.^2+Gy.^2+By.^2;
gxy=Rx.*Ry+Gx.*Gy+Bx.*By;
A=0.5*(atan(2*gxy./(gxx-gyy+eps)));
G1=0.5*((gxx+gyy)+(gxx-gyy).*cos(2*A)+2*gxy.*sin(2*A));
%Now repeat for angle+pi/2, then select maximum at each point
A=A+pi/2;
G2=0.5*((gxx+gyy)+(gxx-gyy).*cos(2*A)+2*gxy.*sin(2*A));
G1=G1.^0.5;
G2=G2.^0.5;
%Form VG by picking maximum at each (x,y) and then convert to a range
%of [0,1]
VG=mat2gray(max(G1,G2));
clear Rx;clear Gx;clear Bx;clear Ry;clear Gy;clear By;
clear gxx;clear gxy;clear gyy;clear G1;clear G2;
   VG1=imfill(VG);                         %Filling the connected edges
for j=1:rows                          %Thresholding the image at VG1>0.1 and converting to binary
   for k=1:cols
      if(VG1(j,k)>0.1)
         BWB(j,k)=1;
      else
         BWB(j,k)=0;
      end;
   end;
end;
clear VG1;
```

```
%-------flash region found----------------------
  SE=strel('disk',2);                 %Dilating by 8 pixels to obtain neighborhood regions
  Flash_Surr_Region=imdilate(BW2,SE);
%    imshow(uint8(Flash_Surr_Region));
  temp_BW2 = BW2;
  SE=strel('disk',1);                 %Dilating it by 4 pixels to obtain the actual flash region
  BW2=imdilate(BW2,SE);
%Iterative Region Smoothing
[Conn num]=bwlabel(Flash_Surr_Region);          %Obtains the labeled regions of the dilated flash
neighborhood region
  [D,L]=bwdist(BW2);                  %Finds distance of each zero pixel to nearest non zero pixel
  dir=zeros(1,num);                   %Holds the direction for each connected region
  [rows cols] = size(Conn);
  for main=1:num                      %Loops through each connected region
   totalgrad=0;
   cnt=0;
   [x,y]=find(Conn==main);             %Find all pixels of a particular connected region
   for loop=1:length(x);              %and loop through them
         j=x(loop);
         k=y(loop);
         if (j >= 1 and j <= rows and k >= 1 and k <= cols)
           flag=0;
           for m=j-1:j+1              %Look at a 3x3 neighborhood of each pixel in the region
             for n=k-1:k+1            %and make flag 1 if one of its neighbors lies outside flash region
               if (m >=1 and m <= rows and n >= 1 and n <= cols)
                 if(Conn(m,n)==0)     %i.e, if the pixel lies in the border
                   flag=1;
                 end;
               end;
             end;
           end;
           if(flag==1)
             posy=ceil(L(j,k)/cols); %rows);   %obtains position of the nearest flash pixel
```

```
                posx=mod(L(j,k),rows);
                if (posx >= 1 and posx <= rows and posy >= 1 and posy <= cols)
                   dir(main)=dir(main)+(A(j,k)*VG(posx,posy));   %Holds the sum of weighted directions of each such pixel
                   totalgrad=totalgrad+VG(posx,posy);        %Holds the sum of gradients
                   cnt=cnt+1;
                end;
             end;
          end;
   end;
   clear j
   if(dir(main)~=0)
      dir(main)=dir(main)/(cnt*totalgrad);   %Obtaining the weighted mean for direction for each connected region
   end;
  end;
  clear j
  [xcount,ycount]=find(BW2>0);            %Finds the number of flash pixels in flash mask
  cntWtBar = 0;
  while(length(xcount)~=0)                %While this number is not equal to zero loop
    cntWtBar = cntWtBar + 1;
    waitbar(cntWtBar/length(xcount))
     newcounter=1;
     for j=1:length(xcount)
        flag=0;
        for m=xcount(j)-1:xcount(j)+1      %Checks for border pixels
           for n=ycount(j)-1:ycount(j)+1
             if m > 1 and m < a and n > 1 and n < b   %to avoid null error for pixel at image border
               if(BW2(m,n)==0)
                  flag=1;
               end;
             end
           end;
```

```
      end;
      v = 0;
      if(flag==1)
        borderregion(xcount(j),ycount(j))=1; %Creates a border mask
        xcount_1(newcounter)=xcount(j);
        ycount_1(newcounter)=ycount(j);
        newcounter=newcounter+1;
      end;
    end;
    for j=1:length(xcount_1)                  %Loops for each pixel in the bordermask
      curr_dir=dir(Conn(xcount_1(j),ycount_1(j)));    %Holds the direction of smoothing for the border
pixel based on the label
      counter=1;
      flag=0;
      total=0;
      for m=xcount_1(j)-5:xcount_1(j)+5          %Loop through a 10x10 neighborhood
        for n=ycount_1(j)-5:ycount_1(j)+5
          if m > 5 and m < a-5 and n > 5 and n < b-5
           if(BW2(m,n)==0)
              posy=ceil(L(m,n)/cols); %rows);          %obtains position of the nearest flash pixel
              posx=mod(L(m,n),rows);
       R(counter)=colImage(m,n,1)*VG(posx,posy); %Holds sum of weighted gradient at that position
              G(counter)=colImage(m,n,2)*VG(posx,posy);
              B(counter)=colImage(m,n,3)*VG(posx,posy);
              total=total+(VG(posx,posy));
              counter=counter+1;
              flag=1;
           end;
          end
        end;
      end;
      counter=counter-1;
      if(flag==1)                         %Calculates the weighted mean color for smoothing
```

```
    red=0;blue=0;green=0;
    for main=1:counter
          red=red+R(main);
          blue=blue+B(main);
          green=green+G(main);
    end;
    if(total~=0)
       red=red/total;
       blue=blue/total;
       green=green/total;
    end;
end;
% Substituting the color value in the correct pixel based on
% direction provided the pixel in the appropriate direction is
% used in calculation of pixel value
    if xcount_1(j) > 1 and xcount_1(j) < a and ycount_1(j) > 1 and ycount_1(j) < b
if (curr_dir>5.8905 | curr_dir<=0.3925)
    if(BW2(xcount_1(j),ycount_1(j)+1)==0)
       colImage(xcount_1(j),ycount_1(j),1)=red;
       colImage(xcount_1(j),ycount_1(j),2)=green;
       colImage(xcount_1(j),ycount_1(j),3)=blue;
       BW2(xcount_1(j),ycount_1(j))=0;
    end;
elseif (curr_dir>0.39255 and curr_dir<=1.1775)
    if(BW2(xcount_1(j)+1,ycount_1(j)+1)==0)
       colImage(xcount_1(j),ycount_1(j),1)=red;
       colImage(xcount_1(j),ycount_1(j),2)=green;
       colImage(xcount_1(j),ycount_1(j),3)=blue;
       BW2(xcount_1(j),ycount_1(j))=0;
    end;
elseif (curr_dir>1.1775 and curr_dir<=1.9625)
    if(BW2(xcount_1(j)+1,ycount_1(j))==0)
       colImage(xcount_1(j),ycount_1(j),1)=red;
```

```
        colImage(xcount_1(j),ycount_1(j),2)=green;
        colImage(xcount_1(j),ycount_1(j),3)=blue;
        BW2(xcount_1(j),ycount_1(j))=0;
    end;
elseif (curr_dir>1.9625 and curr_dir<=2.7489)
    if(BW2(xcount_1(j)+1,ycount_1(j)-1)==0)
        colImage(xcount_1(j),ycount_1(j),1)=red;
        colImage(xcount_1(j),ycount_1(j),2)=green;
        colImage(xcount_1(j),ycount_1(j),3)=blue;
        BW2(xcount_1(j),ycount_1(j))=0;
    end;
elseif (curr_dir>2.7489 and curr_dir<=3.5343)
    if(BW2(xcount_1(j),ycount_1(j)-1)==0)
        colImage(xcount_1(j),ycount_1(j),1)=red;
        colImage(xcount_1(j),ycount_1(j),2)=green;
        colImage(xcount_1(j),ycount_1(j),3)=blue;
        BW2(xcount_1(j),ycount_1(j))=0;
    end;
elseif (curr_dir>3.5343 and curr_dir<=4.3197)
    if(BW2(xcount_1(j)-1,ycount_1(j)-1)==0)
        colImage(xcount_1(j),ycount_1(j),1)=red;
        colImage(xcount_1(j),ycount_1(j),2)=green;
        colImage(xcount_1(j),ycount_1(j),3)=blue;
        BW2(xcount_1(j),ycount_1(j))=0;
    end;
elseif (curr_dir>4.3197 and curr_dir<=5.1051)
    if(BW2(xcount_1(j)-1,ycount_1(j))==0)
        colImage(xcount_1(j),ycount_1(j),1)=red;
        colImage(xcount_1(j),ycount_1(j),2)=green;
        colImage(xcount_1(j),ycount_1(j),3)=blue;
        BW2(xcount_1(j),ycount_1(j))=0;
    end;
elseif (curr_dir>5.1051 and curr_dir<=5.8905)
```

```
            if(BW2(xcount_1(j)-1,ycount_1(j)+1)==0)
               colImage(xcount_1(j),ycount_1(j),1)=red;
               colImage(xcount_1(j),ycount_1(j),2)=green;
               colImage(xcount_1(j),ycount_1(j),3)=blue;
               BW2(xcount_1(j),ycount_1(j))=0;
            end;
         end;
      end
         clear R;clear G;clear B;
      end;
      clear xcount_1;clear ycount_1;
      [xcount,ycount]=find(BW2>0);
   end;
   clear xcount ycount
%----applying local filter instead of foltering entire image
   [xcount,ycount]=find(temp_BW2>0);
   for k = 1:length(xcount)
     if xcount(k) > 2 and xcount(k) < a-2 and ycount(k) > 2 and ycount(k) < b-2
       temp = colImage(xcount(k)-2:xcount(k)+2,ycount(k)-2:ycount(k)+2,1:3);
       colImage(xcount(k),ycount(k),1) = mean(mean(temp(:,:,1)));
       colImage(xcount(k),ycount(k),2) = mean(mean(temp(:,:,2)));
       colImage(xcount(k),ycount(k),3) = mean(mean(temp(:,:,3)));
     end
   end
   close(h)
   nonFlashImg = colImage(6:(a-5),6:(b-5),1:3);
```

APPENDIX F

SCORING RESULTS OF FLASH REMOVAL ON
TRAINING SET OF 85 IMAGES

Table F1 Scoring of flash removed images for training set using Fixed Number Method

Image	Fix Number Method							
	Pixel Based				Region Based			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Abl108	65.5584	1.5967	68.3038	0.0492	79.0323	331	72.4138	7
Abl161	19.4030	10.6374	0.0000	0.0000	14.2857	1588	0.0000	0
Abl195	44.0000	2.9416		0.7291	40.0000	884		409
Acl205	33.7294	1.0152	11.1724	0.0297	63.2118	478	30.3448	17
Acl221	86.8293	4.5198		0.0090	90.4762	338		1
Adl313	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Adl337bis	65.0291	2.6316	41.0868	0.1450	81.9444	575	69.0323	84
Adl350	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Adl389	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Ael409		5.3103		0.1733		608		34
Ael419	71.0884	7.1413		0.0000	100.0000	989		0
Ael494	52.6248	2.5234	36.0000	0.3171	55.5556	345	40.5405	21
Ael514	39.8404	0.8941	33.3333	0.1413	47.0588	143	43.7500	20
Ael523	73.7933	8.6707	19.7211	0.0148	84.6000	733	33.3333	1
FAL006	56.1634	2.1104	40.9228	0.0577	76.6730	545	68.2243	13
FAL036	58.2134	0.3719	26.7176	0.0023	82.9352	93	61.5385	0
FDL052	67.7705	1.7123	67.1821	0.1147	86.1386	308	82.6667	30
FFL052	56.7974	1.0826	55.7867	0.3371	88.9262	411	88.7500	200
Fal096	38.9907	0.1487	28.6212	0.0601	49.3939	59	42.1569	39
Fbl030	46.0364	1.6339	34.0456	0.0351	72.7848	202	59.3023	15
Fbl056	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Fbl066	69.4980	0.3329	62.2369	0.1129	74.1573	193	69.7802	72
Fbl072	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Fcl034	29.0500	0.4550	12.0225	0.0189	39.7260	43	24.6032	4
Fcl040	73.1061	8.0371	41.9753	0.0041	68.0412	850	41.1765	1
Fcl048	17.9130	1.7031	15.7091	0.0784	30.3419	161	29.3333	13
Fcl062	35.5851	0.1052	17.2103	0.0049	46.4286	13	45.4545	3
Fdl014	82.8496	0.6009	83.2891	0.2074	77.3585	141	79.2453	31
Fdl020	70.4965	4.0700	30.3030	0.0021	74.3590	484	33.3333	0
Fdl056	56.4583	3.3897	64.1096	3.3706	64.5161	1046	71.4286	1038
Fdl071	31.3188	1.9477	21.5322	0.0878	78.3439	354	67.1429	38
Fel038	20.7274	0.2894	6.9184	0.0126	49.6471	58	17.2932	6
Ffl014	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Ffl026	75.3866	3.6448	75.6605	3.3268	79.6482	1368	79.3814	1313
Ffl096	2.7712	0.4093	2.3024	0.0587	6.1856	113	5.5866	21
Fgl005	25.9892	1.9985	29.3371	1.9057	55.1887	983	65.0558	954
Fgl006	72.9768	0.1626	67.9012	0.0489	85.1240	68	77.7778	39
Fgl053	29.5570	0.3651	21.7979	0.1242	63.2787	176	55.8036	82

Fgl083	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Fhl006	69.9252	1.0787	69.9317	0.6396	77.8281	134	76.0000	44
Fhl030	90.2778	0.0155		0.0000	100.0000	15		0
Fhl060	39.3889	0.0616	44.4444	0.0003	69.7872	24	57.1429	0
Fhl082	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Fil094	26.3609	1.6358	25.7451	0.0267	50.0000	228	47.3684	6
Fll118	66.1209	1.8607	67.1735	0.3048	72.0430	338	69.3122	82
Fml026	39.0833	5.5301	20.7059	0.0859	69.3966	870	53.8462	41
Gbl014	29.1471	2.4466	29.1471	2.3297	55.1724	726	55.1724	715
Gcl003	85.7143	4.7995	57.1429	0.0005	66.6667	564	33.3333	1
Ggl074		0.0015		0.0000		1		0
Nal020	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Nal035	72.7034	8.0498	62.1324	0.0412	73.1481	737	70.4082	34
Nal037	82.4219	1.6054	82.4219	1.4967	92.3077	600	92.3077	570
Nal047	73.5075	7.4066	45.2381	0.0071	80.0000	730	53.8462	1
Nal075	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Nal077	49.8551	2.5165	0.0000	0.0492	54.1667	277	0.0000	15
Nal099	82.6087	2.3327	69.5652	0.1241	81.2500	222	66.6667	19
Nbl011	100.0000	8.1326		0.0000	100.0000	99		0
Nbl031	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Ndl018	57.7784	3.1212	53.7313	0.9457	40.5405	442	100.0000	242
Nel041	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Nel097	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Nfl063	60.6061	5.9445	53.8462	0.4892	69.2308	826	100.0000	240
Nfl074	74.9049	2.9440	74.3590	0.1165	73.3333	340	100.0000	66
Ngl076	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Ngl082	42.3807	2.4647	24.2101	0.6259	67.2414	533	57.1429	200
Ngl084	72.2071	3.1481	74.1279	2.5049	76.1905	617	73.6842	532
Nhl013	69.8591	0.9253	0.0000	0.0000	85.2273	102	0.0000	0
Nil002	51.1312	4.8681	43.6620	0.1318	72.7273	719	75.0000	120
Nil010	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Nil037	29.4964	2.8830	31.6953	0.0264	41.9355	425	33.3333	11
Nil056	34.0637	2.3935	34.4330	1.3099	64.0000	703	62.5000	613
Nil068	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Nil096	58.6549	5.9423	55.5921	0.1238	72.2222	426	70.0000	28
Nil100	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Nil103	70.0789	0.3153	60.0000	0.0344	72.4928	25	53.9326	6
Nll098		0.0000		0.0000		0		0
Nml010	25.3165	6.2242	11.2000	0.0033	54.5455	545	37.5000	2
Nml071	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
Nml079		6.8334		0.1642		887		130
gzl28		0.4640		0.0657		112		39
gzl66		0.0000		0.0000		0		0
nal095	56.4103	3.6786	91.4286	0.4568	56.2500	674	80.0000	199
newl010		8.6950		2.2843		1232		399

newl041	40.9066	2.3736	40.9513	2.1210	66.6667	1091	66.6667	1008
nfl070	36.0000	2.5583	36.0000	1.3252	50.0000	711	50.0000	512
Average	41.7495	2.3024	31.1518	0.3461	51.4068	360.6588	42.3235	122.1294

Table F2 Scoring of flash removed images for training set using Fixed Number Method combined with salient feature masks

Image	Fix Number Method combined with Salient Point							
	Pixel Based				Region Based			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Abl108	65.7844	1.5989	68.3609	0.0492	79.0323	332	72.4138	7
Abl161	19.4030	10.6483	0.0000	0.0000	14.2857	1595	0.0000	0
Abl195	44.0000	3.0041		0.7915	40.0000	1010		535
Acl205	34.8817	1.0845	11.7604	0.0316	64.1230	466	31.0345	18
Acl221	86.8293	4.5260		0.0090	90.4762	341		1
Adl313	36.7347	4.6093	7.1429	1.4469	30.0000	532	7.6923	96
Adl337bis	66.2573	2.7092	42.3459	0.1518	83.6111	581	70.3226	84
Adl350	16.7315	1.9244	24.6057	1.6454	11.9403	227	18.9189	172
Adl389	24.3631	1.4276	8.7108	0.7553	52.5000	209	14.2857	109
Ael409		5.3224		0.1744		610		34
Ael419	71.8645	7.1528		0.0000	100.0000	994		0
Ael494	52.7529	2.5296	36.3077	0.3171	55.5556	352	40.5405	21
Ael514	39.9018	0.9130	33.3333	0.1416	47.0588	147	43.7500	20
Ael523	73.8605	8.7080	19.7211	0.0148	84.6000	754	33.3333	1
FAL006	57.5364	2.2022	41.8306	0.0608	77.2467	539	69.1589	15
FAL036	59.4568	0.4129	26.7176	0.0023	83.2765	91	61.5385	0
FDL052	68.5509	1.7373	67.9921	0.1238	86.1386	310	82.6667	32
FFL052	57.3236	1.1287	56.2262	0.3444	89.2617	423	89.3750	203
Fal096	39.3094	0.1529	28.8090	0.0616	49.3939	59	42.1569	40
Fbl030	47.0387	1.6599	34.6154	0.0359	73.4177	207	59.3023	15
Fbl056	6.1086	0.9288	1.4451	0.0683	36.3636	142	20.0000	12
Fbl066	70.7327	0.3787	63.3872	0.1241	74.1573	205	69.7802	75
Fbl072	50.9465	0.6903	7.6923	0.0005	81.0811	165	12.5000	1
Fcl034	29.1742	0.4568	12.0974	0.0189	40.0000	43	25.3968	4
Fcl040	73.4848	8.0760	41.9753	0.0041	68.0412	857	41.1765	1
Fcl048	17.9841	1.7130	15.7584	0.0784	30.3419	164	29.3333	13
Fcl062	35.6409	0.1052	17.2103	0.0049	46.4286	13	45.4545	3
Fdl014	82.8496	0.6062	83.2891	0.2076	77.3585	136	79.2453	31
Fdl020	70.4965	4.0921	30.3030	0.0021	74.3590	489	33.3333	0
Fdl056	56.4583	3.3932	64.1096	3.3738	64.5161	1052	71.4286	1043
Fdl071	31.8208	1.9630	21.9628	0.0891	78.3439	362	67.1429	38
Fel038	20.8480	0.2966	6.9184	0.0126	49.6471	58	17.2932	6
Ffl014	61.5794	0.2692	51.9470	0.0300	70.1863	31	60.1626	2

Ffl026	75.3866	3.6461	75.6605	3.3270	79.6482	1370	79.3814	1314
Ffl096	2.7914	0.4095	2.3169	0.0587	6.1856	114	5.5866	21
Fgl005	25.9892	1.9993	29.3371	1.9065	55.1887	985	65.0558	956
Fgl006	73.6208	0.1671	68.2305	0.0495	85.1240	71	77.7778	39
Fgl053	29.8277	0.3745	22.0143	0.1292	63.6066	178	56.2500	84
Fgl083	28.8260	0.3598	24.3544	0.1229	49.8008	118	44.9438	50
Fhl006	70.1247	1.0826	70.1025	0.6422	77.8281	139	76.0000	46
Fhl030	90.2778	0.0432		0.0000	100.0000	24		0
Fhl060	39.5926	0.0633	44.4444	0.0003	69.7872	25	57.1429	0
Fhl082	53.5879	1.0617	53.7100	0.3077	67.8899	240	61.3636	80
Fil094	26.3609	1.6389	25.7451	0.0267	50.0000	232	47.3684	6
Fll118	66.5357	1.8847	67.7263	0.3109	72.0430	347	69.3122	87
Fml026	39.3662	5.5461	20.7647	0.0862	70.2586	886	55.3846	42
Gbl014	29.2736	2.9092	29.2736	2.7923	55.1724	1124	55.1724	1113
Gcl003	85.7143	4.8020	57.1429	0.0005	66.6667	564	33.3333	1
Ggl074		0.0015		0.0000		1		0
Nal020	70.9375	2.9309	60.7143	0.3776	65.6250	302	36.8421	34
Nal035	72.9659	8.0613	62.5000	0.0421	73.1481	741	70.4082	36
Nal037	82.4219	1.6349	82.4219	1.5263	92.3077	642	92.3077	612
Nal047	73.5075	7.4247	45.2381	0.0071	80.0000	740	53.8462	1
Nal075	59.8499	2.9868	32.3810	0.1436	71.4286	384	20.0000	31
Nal077	50.1449	2.5500	0.0000	0.0492	54.1667	273	0.0000	15
Nal099	82.6087	2.3434	69.5652	0.1244	81.2500	224	66.6667	19
Nbl011	100.0000	8.1326		0.0000	100.0000	99		0
Nbl031	70.7448	0.5905	64.7897	0.1207	85.0806	73	77.6596	18
Ndl018	57.7784	3.2891	53.7313	1.0472	40.5405	720	100.0000	430
Nel041	23.1351	1.2333	22.5426	0.0025	40.0000	226	40.0000	0
Nel097	82.7411	3.2072	83.5897	0.4015	58.3333	394	63.6364	58
Nfl063	60.6061	5.9505	53.8462	0.4952	69.2308	843	100.0000	257
Nfl074	74.9049	3.0217	74.3590	0.1550	73.3333	432	100.0000	107
Ngl076	55.3571	5.3396	54.0541	0.1172	62.5000	623	60.0000	11
Ngl082	42.3807	2.4761	24.2101	0.6259	67.2414	539	57.1429	200
Ngl084	72.2071	3.1562	74.1279	2.5130	76.1905	626	73.6842	541
Nhl013	69.9045	0.9520	0.0000	0.0000	85.2273	115	0.0000	0
Nil002	52.0362	4.8812	45.0704	0.1450	72.7273	729	75.0000	130
Nil010	65.0000	4.4357	94.1176	0.0595	75.0000	498	100.0000	13
Nil037	29.4964	2.8841	31.6953	0.0264	41.9355	426	33.3333	11
Nil056	34.0637	2.4160	34.4330	1.3314	64.0000	745	62.5000	652
Nil068	33.3760	1.2588	47.4359	0.0911	66.6667	213	52.1739	17
Nil096	59.0959	5.9565	55.5921	0.1246	72.2222	426	70.0000	28
Nil100	13.8588	0.6380	7.1374	0.1396	23.5602	110	13.2075	21
Nil103	70.2205	0.3183	60.0000	0.0344	72.4928	25	53.9326	6
Nll098		4.8603		0.0345		616		11
Nml010	25.3165	6.2516	11.2000	0.0033	54.5455	550	37.5000	2
Nml071	62.7626	1.1033	83.3333	0.2973	84.2105	146	66.6667	31

Nml079		6.8451		0.1760		908		151
gzl28		0.4686		0.0703		122		49
gzl66		0.0000		0.0000		0		0
nal095	56.4103	3.7487	91.4286	0.4942	56.2500	691	80.0000	213
newl010		8.9271		2.5163		1407		574
newl041	40.9066	2.3911	40.9513	2.1350	66.6667	1102	66.6667	1017
nfl070	36.0000	2.6870	36.0000	1.3716	50.0000	809	50.0000	600
Average	52.4195	2.7972	41.2858	0.4321	64.7157	440.39	52.9587	145.4941

Table F3 Scoring of flash removed images for training set using Standard Deviation Method

Image	Standard Deviation Method							
	Pixel Based				Region Based			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Abl108	65.5584	1.5967	68.3038	0.0492	79.0323	331	72.4138	7
Abl161	19.4030	10.6374	0.0000	0.0000	14.2857	1588	0.0000	0
Abl195	44.0000	2.9416		0.7291	40.0000	884		409
Acl205	33.7294	1.0152	11.1724	0.0297	63.2118	478	30.3448	17
Acl221	86.8293	4.5198		0.0090	90.4762	338		1
Adl313	36.7347	4.6093	7.1429	1.4469	30.0000	532	7.6923	96
Adl337bis	65.0291	2.6316	41.0868	0.1450	81.9444	575	69.0323	84
Adl350	16.7315	1.9244	24.6057	1.6454	11.9403	227	18.9189	172
Adl389	24.3631	1.4276	8.7108	0.7553	52.5000	209	14.2857	109
Ael409		5.3103		0.1733		608		34
Ael419	71.0884	7.1413		0.0000	100.0000	989		0
Ael494	52.6248	2.5234	36.0000	0.3171	55.5556	345	40.5405	21
Ael514	39.8404	0.8941	33.3333	0.1413	47.0588	143	43.7500	20
Ael523	73.7933	8.6707	19.7211	0.0148	84.6000	733	33.3333	1
FAL006	56.1634	2.1104	40.9228	0.0577	76.6730	545	68.2243	13
FAL036	58.2134	0.3719	26.7176	0.0023	82.9352	93	61.5385	0
FDL052	67.7705	1.7123	67.1821	0.1147	86.1386	308	82.6667	30
FFL052	56.7974	1.0826	55.7867	0.3371	88.9262	411	88.7500	200
Fal096	38.9907	0.1487	28.6212	0.0601	49.3939	59	42.1569	39
Fbl030	46.0364	1.6339	34.0456	0.0351	72.7848	202	59.3023	15
Fbl056	6.1086	0.9288	1.4451	0.0683	36.3636	142	20.0000	12
Fbl066	69.4980	0.3329	62.2369	0.1129	74.1573	193	69.7802	72
Fbl072	50.9465	0.6823	7.6923	0.0005	81.0811	158	12.5000	1
Fcl034	29.0500	0.4550	12.0225	0.0189	39.7260	43	24.6032	4
Fcl040	73.1061	8.0371	41.9753	0.0041	68.0412	850	41.1765	1
Fcl048	17.9130	1.7031	15.7091	0.0784	30.3419	161	29.3333	13
Fcl062	35.5851	0.1052	17.2103	0.0049	46.4286	13	45.4545	3
Fdl014	82.8496	0.6009	83.2891	0.2074	77.3585	141	79.2453	31
Fdl020	70.4965	4.0700	30.3030	0.0021	74.3590	484	33.3333	0
Fdl056	56.4583	3.3897	64.1096	3.3706	64.5161	1046	71.4286	1038

Fdl071	31.3188	1.9477	21.5322	0.0878	78.3439	354	67.1429	38
Fel038	20.7274	0.2894	6.9184	0.0126	49.6471	58	17.2932	6
Ffl014	61.4907	0.2689	51.9470	0.0300	70.1863	31	60.1626	2
Ffl026	75.3866	3.6448	75.6605	3.3268	79.6482	1368	79.3814	1313
Ffl096	2.7712	0.4093	2.3024	0.0587	6.1856	113	5.5866	21
Fgl005	25.9892	1.9985	29.3371	1.9057	55.1887	983	65.0558	954
Fgl006	72.9768	0.1626	67.9012	0.0489	85.1240	68	77.7778	39
Fgl053	29.5570	0.3651	21.7979	0.1242	63.2787	176	55.8036	82
Fgl083	28.8178	0.3595	24.3544	0.1229	49.8008	118	44.9438	50
Fhl006	69.9252	1.0787	69.9317	0.6396	77.8281	134	76.0000	44
Fhl030	90.2778	0.0155		0.0000	100.0000	15		0
Fhl060	39.3889	0.0616	44.4444	0.0003	69.7872	24	57.1429	0
Fhl082	53.5419	1.0449	53.7100	0.3051	67.8899	230	61.3636	77
Fil094	26.3609	1.6358	25.7451	0.0267	50.0000	228	47.3684	6
Fll118	66.1209	1.8607	67.1735	0.3048	72.0430	338	69.3122	82
Fml026	39.0833	5.5301	20.7059	0.0859	69.3966	870	53.8462	41
Gbl014	29.1471	2.4466	29.1471	2.3297	55.1724	726	55.1724	715
Gcl003	85.7143	4.7995	57.1429	0.0005	66.6667	564	33.3333	1
Ggl074		0.0015		0.0000		1		0
Nal020	70.9375	2.9309	60.7143	0.3776	65.6250	302	36.8421	34
Nal035	72.7034	8.0498	62.1324	0.0412	73.1481	737	70.4082	34
Nal037	82.4219	1.6054	82.4219	1.4967	92.3077	600	92.3077	570
Nal047	73.5075	7.4066	45.2381	0.0071	80.0000	730	53.8462	1
Nal075	59.8499	2.9868	32.3810	0.1436	71.4286	384	20.0000	31
Nal077	49.8551	2.5165	0.0000	0.0492	54.1667	277	0.0000	15
Nal099	82.6087	2.3327	69.5652	0.1241	81.2500	222	66.6667	19
Nbl011	100.0000	8.1326		0.0000	100.0000	99		0
Nbl031	70.7448	0.5905	64.7897	0.1207	85.0806	73	77.6596	18
Ndl018	57.7784	3.1212	53.7313	0.9457	40.5405	442	100.0000	242
Nel041	23.1351	1.2333	22.5426	0.0025	40.0000	226	40.0000	0
Nel097	82.7411	3.1772	83.5897	0.4013	58.3333	387	63.6364	57
Nfl063	60.6061	5.9445	53.8462	0.4892	69.2308	826	100.0000	240
Nfl074	74.9049	2.9440	74.3590	0.1165	73.3333	340	100.0000	66
Ngl076	55.3571	5.3396	54.0541	0.1172	62.5000	623	60.0000	11
Ngl082	42.3807	2.4647	24.2101	0.6259	67.2414	533	57.1429	200
Ngl084	72.2071	3.1481	74.1279	2.5049	76.1905	617	73.6842	532
Nhl013	69.8591	0.9253	0.0000	0.0000	85.2273	102	0.0000	0
Nil002	51.1312	4.8681	43.6620	0.1318	72.7273	719	75.0000	120
Nil010	65.0000	4.4357	94.1176	0.0595	75.0000	498	100.0000	13
Nil037	29.4964	2.8830	31.6953	0.0264	41.9355	425	33.3333	11
Nil056	34.0637	2.3935	34.4330	1.3099	64.0000	703	62.5000	613
Nil068	33.3760	1.2588	47.4359	0.0911	66.6667	213	52.1739	17
Nil096	58.6549	5.9423	55.5921	0.1238	72.2222	426	70.0000	28
Nil100	13.8588	0.6380	7.1374	0.1396	23.5602	110	13.2075	21
Nil103	70.0789	0.3153	60.0000	0.0344	72.4928	25	53.9326	6
Nll098		4.8603		0.0345		616		11
Nml010	25.3165	6.2242	11.2000	0.0033	54.5455	545	37.5000	2
Nml071	62.7626	1.1033	83.3333	0.2973	84.2105	146	66.6667	31

Nml079		6.8334		0.1642		887		130
gzl28		0.4640		0.0657		112		39
gzl66		0.0000		0.0000		0		0
nal095	56.4103	3.6786	91.4286	0.4568	56.2500	674	80.0000	199
newl010		8.6950		2.2843		1232		399
newl041	40.9066	2.3736	40.9513	2.1210	66.6667	1091	66.6667	1008
nfl070	36.0000	2.5583	36.0000	1.3252	50.0000	711	50.0000	512
Average	52.2174	2.7706	41.1478	0.4185	64.6397	422.1294	52.8721	131.1059

Table F4 Scoring of flash removed images for training set using Standard Deviation Method combined with salient feature masks

Image	Standard Deviation Method combined with Salient Point							
	Pixel Based				Region Based			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Abl108	59.6514	2.2916	73.8435	1.0062	82.2581	466	86.2069	181
Abl161	23.8806	6.9609	23.0769	0.1111	28.5714	1280	33.3333	43
Abl195	44.0000	3.3077		1.0951	40.0000	963		488
Acl205	20.4835	1.1625	13.0834	0.8470	43.8497	415	35.8621	322
Acl221	42.4390	3.3458		0.1975	52.3810	398		110
Adl313	39.7959	5.4146	17.8571	2.2516	36.6667	750	23.0769	313
Adl337bis	45.1034	2.3078	25.3810	1.9838	47.2222	684	34.1935	640
Adl350	58.7549	4.8085	60.2524	3.3768	65.6716	990	72.9730	646
Adl389	25.7962	2.5408	14.9826	1.5401	57.5000	619	28.5714	404
Ael409		3.8280		0.4051		1040		107
Ael419	60.9279	0.2580		0.0222	100.0000	29		19
Ael494	63.7644	1.5145	92.0000	0.3063	75.0000	256	89.1892	95
Ael514	57.8883	2.1050	56.9892	1.4086	64.7059	485	62.5000	364
Ael523	50.8301	1.8116	10.7570	0.1940	64.2000	259	22.2222	112
FAL006	41.0949	1.5187	56.2784	0.8222	50.2868	298	71.0280	180
FAL036	40.6414	0.7228	2.2901	0.0566	48.1229	314	7.6923	74
FDL052	70.7235	3.0747	73.4904	2.9794	76.2376	953	82.6667	925
FFL052	57.2735	4.2306	70.0850	2.7101	79.5302	1281	87.5000	852
Fal096	50.8499	2.8865	58.1277	2.8057	70.6061	952	77.9412	933
Fbl030	41.7540	2.6847	45.0855	1.4969	75.3165	664	70.9302	487
Fbl056	61.7647	1.6162	72.5434	0.7557	63.6364	392	80.0000	262
Fbl066	50.0136	2.4269	56.4856	2.3866	52.1961	884	59.7527	884
Fbl072	51.9488	1.6138	29.4872	0.8898	83.7838	515	25.0000	348
Fcl034	26.9999	1.2253	39.6067	1.0069	47.9452	509	62.6984	485
Fcl040	61.7424	3.0571	64.1975	0.2216	54.6392	411	67.6471	128
Fcl048	44.1570	3.6334	47.3736	2.3131	73.5043	965	74.6667	835
Fcl062	33.2180	1.2658	24.8594	1.1693	51.1905	502	50.0000	482
Fdl014	86.5435	1.6428	86.7374	1.4080	81.1321	714	81.1321	609
Fdl020	60.1418	2.6469	45.4545	0.1583	66.6667	399	66.6667	96
Fdl056	61.8750	2.3379	71.7808	2.3238	67.7419	925	76.1905	919

Fdl071	33.5007	1.8903	35.2221	1.7400	70.7006	566	71.4286	509
Fel038	17.5868	0.3673	7.0865	0.2326	46.5882	173	18.7970	132
Ffl014	65.9494	1.6979	68.2684	1.4237	78.2609	547	81.3008	507
Ffl026	81.1211	3.6559	81.4729	3.3572	86.1809	1261	86.0825	1198
Ffl096	27.5913	6.5752	29.0076	6.0988	63.4021	1747	66.4804	1617
Fgl005	28.7944	2.2145	33.4605	2.1312	59.6698	983	71.7472	943
Fgl006	70.2604	1.4135	75.8025	1.2356	85.6749	808	85.3535	766
Fgl053	32.0205	1.8738	34.7252	1.6894	68.1967	729	71.4286	673
Fgl083	35.2694	1.7602	35.9368	1.4675	60.1594	1042	57.3034	941
Fhl006	76.4090	2.5807	79.8405	2.1796	87.3303	689	87.5000	593
Fhl030	77.7778	0.7112		0.7065	83.3333	370		368
Fhl060	42.1770	4.7941	57.7778	1.6427	80.4255	1464	64.2857	620
Fhl082	69.3652	5.3646	77.1273	4.5768	92.6606	1736	92.0455	1571
Fil094	57.4924	2.5400	58.0216	1.2748	91.6667	664	91.2281	444
Fll118	70.3340	1.6943	77.1942	1.4561	81.0753	751	83.0688	703
Fml026	34.8109	2.6175	35.8824	0.8351	61.6379	651	66.1538	330
Gbl014	34.2609	2.3707	34.2609	2.2738	70.1149	851	70.1149	828
Gcl003	90.4762	5.8701	71.4286	2.2485	83.3333	1293	66.6667	759
Ggl074		0.9018		0.7494		431		322
Nal020	73.4375	3.2281	75.0000	0.6541	75.0000	458	68.4211	172
Nal035	94.2257	3.8263	95.2206	1.1333	98.1481	825	97.9592	487
Nal037	81.8359	2.0132	81.8359	1.8945	92.3077	652	92.3077	623
Nal047	71.2687	3.7927	83.3333	0.3612	74.5455	500	76.9231	144
Nal075	66.6041	4.0437	66.6667	1.2004	81.4286	668	66.6667	315
Nal077	55.0725	1.7852	70.6667	0.2545	66.6667	324	44.4444	134
Nal099	67.2464	1.7217	55.6522	0.6078	81.2500	429	66.6667	258
Nbl011	100.0000	2.0351		0.1260	100.0000	39		6
Nbl031	74.6314	3.1541	77.0492	1.5695	90.3226	1035	88.2979	592
Ndl018	57.7784	2.9001	53.7313	0.7513	40.5405	410	100.0000	215
Nel041	39.6757	2.5376	38.2700	0.5158	65.7143	567	63.3333	156
Nel097	93.4010	3.8955	94.3590	1.0439	83.3333	661	90.9091	312
Nfl063	65.1515	5.9092	76.9231	0.4539	69.2308	774	100.0000	188
Nfl074	74.9049	3.1412	74.3590	0.1157	73.3333	412	100.0000	61
Ngl076	85.7143	5.8451	100.0000	0.5967	87.5000	876	100.0000	251
Ngl082	47.4602	2.4344	25.8689	0.6096	74.1379	655	57.1429	212
Ngl084	74.3869	2.7997	76.4535	2.1437	85.7143	662	84.2105	577
Nhl013	63.3126	1.0092	14.5455	0.7652	84.0909	267	33.3333	255
Nil002	49.7738	4.8664	41.5493	0.1302	72.7273	708	75.0000	109
Nil010	65.0000	4.5850	94.1176	0.2060	75.0000	588	100.0000	100
Nil037	41.7266	2.3566	53.5627	0.8316	58.0645	809	71.4286	527
Nil056	37.0518	2.7547	37.5258	1.6870	68.0000	858	66.6667	767
Nil068	35.2302	2.1645	56.7308	0.9479	72.9167	450	65.2174	246
Nil096	62.9548	4.9166	68.5855	1.4034	88.8889	885	100.0000	511
Nil100	23.0627	2.1111	33.5611	1.6075	38.2199	614	48.4277	517
Nil103	65.7698	2.8007	59.1515	0.4399	76.2178	776	65.1685	195
Nll098		5.6122		0.7861		1045		439
Nml010	25.0000	4.0616	15.2000	0.1332	45.4545	532	37.5000	66
Nml071	62.7626	2.0429	83.3333	1.1806	84.2105	499	66.6667	352

Nml079		6.7924		0.1203		862		102
gzl28		0.4367		0.0385		118		45
gzl66		2.1455		2.1394		1180		1175
nal095	58.4615	4.1294	97.1429	0.6837	62.5000	725	100.0000	213
newl010		8.6237		2.2129		1268		435
newl041	45.0027	2.5995	45.0519	2.3580	71.8750	1234	71.8750	1136
nfl070	36.0000	2.5646	36.0000	1.2742	50.0000	691	50.0000	489
Average	54.7841	2.9503	54.8777	1.2770	69.7220	707.3412	68.6191	453.5176

Table F5 Scoring of flash removed images for training set using Curve fitting Method

Image	Curve Fitting Method							
	Pixel Based				Region Based			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
Abl108	68.0116	2.3526	74.8144	0.6288	88.7097	559	89.6552	157
Abl161	31.3433	7.1290	7.6923	0.0597	42.8571	1401	16.6667	31
Abl195	44.0000	3.9588		0.4006	40.0000	1054		249
Acl205	30.2881	2.2737	47.4458	1.2926	59.6811	675	81.3793	342
Acl221	75.6098	5.6953		0.2296	90.4762	1079		84
Adl313	43.8776	6.2952	32.1429	2.1109	40.0000	670	30.7692	211
Adl337bis	51.9716	2.1836	52.2200	1.7899	64.7222	568	72.9032	510
Adl350	47.4708	8.5554	46.0568	5.3777	46.2687	869	48.6486	497
Adl389	63.7739	3.8361	52.2648	1.3874	75.0000	661	50.0000	210
Ael409		3.2026		0.2811		795		99
Ael419	60.9984	0.6431		0.0088	100.0000	302		12
Ael494	65.8131	3.3358	83.6923	0.1936	84.7222	550	91.8919	65
Ael514	69.1835	2.0421	69.3135	0.9801	96.0784	483	95.8333	322
Ael523	55.6856	3.2813	34.0637	0.0714	74.0000	505	69.4444	64
FAL006	47.9858	3.1500	72.0877	0.5497	73.0402	858	88.7850	169
FAL036	53.7140	4.1579	53.4351	0.0413	80.2048	1791	92.3077	48
FDL052	75.7857	2.0595	78.9396	1.9566	84.1584	645	92.0000	614
FFL052	62.7365	3.1030	75.5640	1.2922	88.9262	816	97.5000	445
Fal096	59.5086	4.1207	64.3777	1.8571	89.6970	1711	93.1373	784
Fbl030	54.0774	2.4491	55.8405	0.6727	87.9747	608	83.7209	279
Fbl056	53.1674	3.9306	54.3353	1.0686	90.9091	868	100.0000	350
Fbl066	69.4980	0.3329	62.2369	0.1129	74.1573	193	69.7802	72
Fbl072	61.0802	3.1983	55.1282	1.0060	94.5946	873	100.0000	399
Fcl034	35.8645	1.9836	45.7116	0.7249	77.8082	910	88.0952	332
Fcl040	73.4848	3.2378	91.3580	0.0771	67.0103	503	91.1765	50
Fcl048	43.1191	2.8638	45.6008	1.0382	73.0769	735	74.2222	455
Fcl062	44.4395	3.6587	40.9449	1.1739	91.6667	1769	86.3636	476
Fdl014	94.7230	1.7583	94.9602	1.4410	96.2264	613	96.2264	468
Fdl020	68.7943	4.2050	46.9697	0.0991	94.8718	875	100.0000	57

Fdl056	89.1667	2.5192	96.4384	1.9834	93.5484	527	95.2381	427
Fdl071	38.9168	1.4915	42.4751	0.7064	80.2548	549	84.2857	311
Fel038	30.2422	2.0450	26.6908	0.9466	71.7647	1296	69.9248	547
Ffl014	76.5306	2.7975	86.8268	1.2548	94.7205	1103	99.1870	506
Ffl026	88.2732	1.9805	88.8045	1.6001	95.4774	687	95.6186	632
Ffl096	27.5644	3.1998	28.4590	2.8379	69.5876	1115	72.0670	980
Fgl005	41.4779	2.1479	44.2960	1.2361	90.5660	1137	92.5651	644
Fgl006	87.1465	2.4936	91.1934	1.1756	98.8981	1272	98.9899	606
Fgl053	43.2501	1.9924	48.3341	1.2376	92.7869	821	91.0714	573
Fgl083	40.8465	2.1529	43.5749	1.2209	76.0956	1295	71.9101	787
Fhl006	82.8928	2.5523	84.1116	2.0342	93.6652	674	93.0000	537
Fhl030	86.1111	1.2653		0.8084	100.0000	699		461
Fhl060	47.8931	4.5335	81.1111	0.8663	86.8085	1470	85.7143	287
Fhl082	75.8970	4.3672	82.4370	2.0688	100.0000	1500	100.0000	909
Fil094	50.8869	5.9802	50.7926	1.1323	93.3333	1086	92.9825	381
Fll118	76.9701	1.5374	84.6925	0.8190	90.1075	681	91.7989	451
Fml026	41.0921	5.1261	32.6471	0.3239	75.8621	968	76.9231	132
Gbl014	41.1818	1.1463	41.1818	0.9650	67.8161	456	67.8161	435
Gcl003	100.0000	5.3438	100.0000	1.5643	100.0000	1260	100.0000	637
Ggl074		3.6902		0.5269		1185		213
Nal020	95.3125	8.9443	94.6429	0.9488	92.1875	1057	84.2105	111
Nal035	93.9633	3.2480	94.8529	0.5104	99.0741	611	98.9796	245
Nal037	93.3594	3.3661	93.3594	3.0536	100.0000	742	100.0000	680
Nal047	77.6119	4.5955	97.6190	0.1990	85.4545	983	100.0000	124
Nal075	74.1088	6.6636	77.1429	0.7150	92.8571	754	80.0000	144
Nal077	52.4638	3.0777	44.0000	0.2705	62.5000	368	22.2222	76
Nal099	82.0290	3.0156	74.7826	0.4277	87.5000	602	73.3333	171
Nbl011	100.0000	2.7712		0.1417	100.0000	59		14
Nbl031	92.0161	4.7699	93.5852	0.9167	100.0000	1137	100.0000	316
Ndl018	60.3287	4.4173	62.6866	0.2959	48.6486	617	100.0000	154
Nel041	53.2973	4.0966	55.1769	0.9490	85.7143	1104	86.6667	350
Nel097	98.9848	6.5750	100.0000	1.7980	91.6667	995	100.0000	402
Nfl063	78.7879	6.0130	84.6154	0.1661	92.3077	835	100.0000	66
Nfl074	78.7072	5.2808	74.3590	0.1105	86.6667	694	100.0000	54
Ngl076	92.8571	11.1082	94.5946	0.4494	100.0000	1393	100.0000	152
Ngl082	51.8728	2.0497	27.1722	0.6455	91.3793	473	92.8571	164
Ngl084	77.1117	2.4238	79.0698	1.5489	95.2381	483	94.7368	354
Nhl013	66.4192	2.3247	45.4545	0.1234	92.0455	626	66.6667	98
Nil002	63.3484	8.3588	47.8873	0.1384	100.0000	1123	100.0000	89
Nil010	67.5000	7.2708	100.0000	0.2210	75.0000	722	100.0000	56
Nil037	57.1942	3.3202	74.6929	0.5498	64.5161	665	80.9524	175
Nil056	79.2829	4.2751	81.2371	2.9582	96.0000	758	95.8333	636
Nil068	42.0716	3.6589	83.3333	0.9635	87.5000	633	91.3043	213
Nil096	70.2315	8.6848	69.7368	1.0740	100.0000	1283	100.0000	349
Nil100	42.7929	4.1379	36.2187	1.4742	69.8953	958	64.1509	415

Nil103	89.9656	2.9809	87.5152	0.5538	97.9943	706	94.3820	166
Nll098		8.0032		0.4056		1300		214
Nml010	39.8734	5.5628	33.6000	0.1487	81.8182	995	87.5000	56
Nml071	77.5302	4.6252	87.5000	0.9448	100.0000	930	83.3333	199
Nml079		7.2636		0.1203		994		58
gzl28		0.5262		0.0754		139		44
gzl66		3.5187		1.9822		1967		1219
nal095	55.3846	5.3628	70.0000	0.2959	68.7500	765	100.0000	93
newl010		8.5653		2.1448		1235		390
newl041	72.3102	2.7223	72.2799	2.4197	83.3333	986	83.3333	893
nfl070	48.0000	3.3834	48.0000	1.3542	75.0000	715	75.0000	340
Average	64.0649	3.9332	65.4298	0.9921	83.8613	872.0824	85.4118	324.5529

APPENDIX G

MULTIPLIER REDUCTION ALGORITHM BY CHOWDHURY-WRONKIEWICZ

```
//Function DermBrighten changes the RGB values of an image to better reflect EDRA set
//input: pSrc (image needing to be changed)
//output: pDst (the resulting image passed by reference)
void DermBrighten(IplImage *pSrc, IplImage *pDst)
{
	//multiplier used that has been determined comparing EDRA set to CNP images
	double multMod[3] = {1.33281, 1.46642, 1.39156};
	unsigned long oversat=0; //counter of oversaturated pixels
	double high[3] = {0}; //highest pixel values
	CvScalar temp;//holds color information temporarily
	//loops used to determine if image is too bright
	for(int m = 0; m<pSrc->height; m++)
	{
		for(int n = 0; n<pSrc->width; n++)
		{
			temp = cvGet2D(pSrc,m,n);
			for(int q = 0; q<3; q++)
			{
//if the pixel being sampled has a value over 255 in the Red, Green, or Blue plane the counter is
incremented
				if(temp.val[q]*mult[q]>255)
					oversat++;
//the highest value in each of the Red, Green, and Blue plane is tracked to update multiplier when image
is oversaturated
				if(temp.val[q]*mult[q]>high[q])
					high[q]=temp.val[q]*mult[q];
			}
		}
	}
	double highest;
	int index;
//if over 390000 pixels are oversaturated the multiplier is changed
//the highest Red Green or Blue value is found and stored in highest
```

```
if(oversat>=390000)
{
        //blue checked here
        if(high[0]>high[1]&&high[0]>high[2])
        {
                highest = high[0]/mult[0];
                index = 0;
        }
        //green checked here
        else if(high[1]>high[0]&&high[1]>high[2])
        {
                highest = high[1]/mult[1];
                index = 1;
        }
        //red checked here
        else
        {
                highest = high[2]/mult[2];
                index = 2;
        }
//the new multiplier is created based on the highest value present in the Red Green and Blue planes
        for(int indexCt = 0; indexCt<3; indexCt++)
                multMod[indexCt] = mult[indexCt]/mult[index]*(255.0/highest);
}
//if the multiplier is ok multmod[] becomes mult[]
else
{
        for(int l = 0; l<3; l++)
                multMod[l] = mult[l];
}
//cvscalar for new pixel
CvScalar newPixel;
//for loop to handle RGB values
```

```
        //for loops to run through every pixel
        for(int i = 0; i<pSrc->height; i++)
        {
                for(int j=0; j<pSrc->width; j++)
                {
                        //pixel is taken from original image
                        newPixel = cvGet2D(pSrc,i,j);
//RGB values are multiplied by the multiplier if there was oversaturation
                        for(int p = 0; p<3; p++)
                                newPixel.val[p]*=multMod[p];
                        //pixel value is set in the new image
                        cvSet2D(pDst,i,j,newPixel);
                }
        }
}
```

APPENDIX H

SCORING RESULTS OF FLASH REMOVAL ON TEST SET OF 99 IMAGES

Table G1 Scoring of flash removed images for test set using Fixed Number Method

Image	Fix Number Method combined with Salient Point							
	Pixel Based				Region Based			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
DSC00189		5.3771		1.4095		5072		3192
aa021907CNP		0.0001		0.0001		1		1
ab021407CNPI	59.8654	4.5076	61.0377	3.4245	81.51261	3013	82.52427	2333
ad021607CNP		0.0576		0.0572		92		90
ag021607CNP		1.2378		0.8363		645		432
ar031607rab67CNP		0.0003		0.0003		1		1
ar031607rab68CNP		0.0003		0.0000		1		0
as030107CNP	4.8112	1.9685	3.4369	1.1734	12	5119	11.36364	3450
bj032907CNP	76.0252	3.2416	50.9804	2.9410	100	1912	100	1648
bm030607cnp	8.9001	0.0124	2.7344	0.0009	37.03704	12	20	2
bs031407CNP		5.6662		3.1425		2509		1790
ca022807CNP	6.7594	1.2535	8.2927	1.2535	36.36364	803	40	803
cd030907CNP2		2.3952		2.3119		1363		1300
ch041907CNP	14.3236	0.4190	14.3236	0.4187	37.5	48	37.5	46
cm031507CNP	16.2791	2.2758	12.7781	1.4579	50	3020	30.76923	2032
cs021507CNP	0.0000	0.0002	0.0000	0.0002	0	1	0	1
cw021907CNP	0.1519	0.0000		0.0000	0	0		0
db021207CNP	2.3697	0.0008		0.0000	25	0		0
db032607CNP	21.2219	2.8442		0.8163	57.14286	2229		830
dk031607CNP	52.9025	2.0569	54.2533	0.6315	88.46154	1413	93.75	189
dl022307CNP	83.1715	0.7404	86.8966	0.7013	100	611	100	556
dp042007CNP	35.0290	2.2309		0.8986	53.90625	1049		302
dr031207CNP	16.7516	2.0873	23.0976	1.5617	35.71429	2006	37.5	1516
ea121506CNP	56.4340	0.6694	57.5770	0.5234	84.61538	303	84.61538	236
eb031907rab69CNP		2.7521		2.1937		2333		1988
eb031907rab70CNP		2.6463		2.2833		1204		1049
eb031907rab71CNP		3.8311		3.6202		1596		1466
eb032607CNP		0.4515		0.0416		840		105
ee040507CNP		0.1410		0.1410		700		700
er031307CNP		0.0001		0.0001		1		1
es022607CNP		3.2885		2.0338		1819		952
ew022807CNP		1.0387		0.7138		797		558
gh021507CNP		2.9568		2.2125		1784		1389
gs031207CNPa	35.5659	0.7375		0.0807	91.66667	1271		155
gs031207CNPb		3.1483		1.9726		2298		1171
gt030907CNP		3.1195		2.8028		2652		2434
hh031207CNPa	7.6555	0.8112	7.6555	0.0003	14.28571	383	14.28571	0

hh031207CNPb	0.0000	0.7177	0.0000	0.6800	0	828	0	823
hh031207CNPc		1.4458		0.0000		713		0
hh031207CNPd	91.2725	1.7754	91.2725	1.7187	95	824	95	762
hs012907CNP		4.7318		2.9149		2590		1554
hs022707CNP		0.0000		0.0000		0		0
im011707CNP		2.4381		2.3078		1596		1368
is021207CNP	23.4270	0.0999	33.4917	0.0558	52.83019	120	76.47059	47
jd012607CNP		1.3545		1.0832		318		270
jd032607CNP		0.6345		0.2437		241		85
jh022307CNP		0.0006		0.0000		1		0
jl021607CNP	2.9657	2.6201	3.1230	1.6493	10	662	11.36364	565
jm021207CNP2		2.8077		2.5189		1318		1105
jm021207CNP	0.0000	0.0000	0.0000	0.0000	0	0	0	0
jm031207CNP		0.2540		0.2300		608		450
jp030907rab55CNP		0.0002		0.0000		1		0
jp030907rab56CNP		0.2306		0.2306		604		604
jp030907rab57CNP		0.9074		0.9074		813		813
jv013007CNPa		2.2005		1.1544		1377		997
ka022707CNP		1.0281		0.5537		1915		991
kc030507rab52CNP		0.0000		0.0000		0		0
kc030507rab53CNP	0.2424	0.0001	9.0909	0.0001	0	0	0	0
kg012607CNP		0.0007		0.0005		2		1
km010807CNP		0.0000		0.0000		0		0
kp012907CNP	10.2616	0.1295	11.7371	0.1150	29.62963	265	25	242
kr030807CNP		0.1424		0.0034		135		3
kr042007CNP		1.1957		0.6466		1354		600
ks021207CNP		0.0000		0.0000		0		0
lf031907CNP		0.6925		0.1252		304		95
ll040307CNP		1.8502		0.1662		794		193
ls012607CNP	0.3797	0.3313	60.0000	0.3313	2.941176	610	100	610
lt030207CNP	2.2381	0.0004	1.4049	0.0000	0	0	0	0
md033007CNP	0.1239	0.0080	0.1239	0.0000	0	2	0	0
mj020907CNP		0.0000		0.0000		0		0
mm012607CNP	0.0587	0.0003	0.0000	0.0000	0.492611	1	0	0
mv021607CNP	0.0000	0.0004	0.0000	0.0000	0	1	0	0
ng021207CNP		3.4904		2.2276		2188		1732
pl032907CNP	27.7034	0.8206		0.1871	50	2188		603
pp022207CNP		1.4008		0.9309		1250		736
rb021207rab24CNP	13.9008	2.2708	13.4328	0.1434	41.81818	1525	40	379
rb021207rab2CNP	38.2255	3.1706	38.2255	1.9797	66.66667	1647	66.66667	1081
rb031207CNP	23.5498	3.8679	30.0132	3.3004	59.01639	2725	64.86486	2459
rm031607CNP		2.3887		0.6531		1440		493
ro030207CNP		1.0408		0.7263		1604		1278
rp041307CNP		0.0357		0.0317		31		26
rr031207CNP		1.7770		1.7258		818		787

se022307CNP		0.2466		0.2466		431		431
sf012907CNP		0.4669		0.4182		169		138
sg021207CNP	31.9420	2.8651	32.5666	2.6835	51.72414	1913	53.84615	1762
sh040307CNP	26.2978	2.1162	26.2978	1.3889	14.28571	1095	14.28571	744
si032307CNP	2.9992	1.5569	6.3642	0.7648	7.964602	962	12.82051	587
ss021207CNP	34.8861	0.1261	20.8333	0.0143	84.21053	184	100	70
ss032007CNP		0.0004		0.0004		2		2
st032107rab72CNP	61.4350	1.9249	61.4350	0.3731	100	1420	100	370
st032107rab73CNP		2.0790		0.7074		1458		529
sw021507CNP	11.1842	0.6116	2.6820	0.1543	42.85714	173	20	32
tb013107CNPb		0.2901		0.2831		118		112
tb022807CNP		0.0021		0.0021		10		10
ts032607CNP		2.6282		1.8144		1925		1348
wc031407CNP		0.0002		0.0000		1		0
wg030107CNP		0.0002		0.0002		1		1
wp020907CNP		0.1974		0.1972		217		215
wr030907CNP		2.1957		0.5334		3175		607
Average	22.5328	1.3044	24.2694	0.8160	40.3661	965.3333	42.1361	620.4848

Table G2 Scoring of flash removed images for test set using Fixed Number Method combined with salient feature masks

Image	Fix Number Method							
	Pixel Based				Region Based			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
DSC00189		3.3563		1.3146		3076		2169
aa021907CNP		0.0001		0.0001		1		1
ab021407CNPl	61.6543	5.5079	63.1091	4.0374	81.51261	3275	83.4951	2475
ad021607CNP		0.0552		0.0549		83		81
ag021607CNP		1.2378		0.8363		645		432
ar031607rab67CNP		0.0005		0.0005		1		1
ar031607rab68CNP		0.0003		0.0000		1		0
as030107CNP	38.4660	5.1735	39.5815	4.3457	64	5978	68.1818	4306
bj032907CNP	83.9117	3.2476	100.0000	2.9480	100	1885	100	1626
bm030607cnp	17.2964	0.3431	3.1250	0.0012	55.55556	131	20	3
bs031407CNP		4.6171		3.1768		2319		1772
ca022807CNP	6.7594	1.3233	8.2927	1.2508	36.36364	809	40	777
cd030907CNP2		2.2313		2.1511		1282		1231
ch041907CNP	28.2493	0.9022	28.2493	0.5650	62.5	278	62.5	189
cm031507CNP	23.9756	2.2965	21.6147	1.4655	71.42857	3022	53.8462	2037
cs021507CNP	0.0000	0.0000	0.0000	0.0000	0	0	0	0
cw021907CNP	0.4811	0.0003		0.0000	0	0		0
db021207CNP	18.8981	0.0125		0.0000	75	4		0
db032607CNP	21.2219	3.2445		1.1801	57.14286	2262		909
dk031607CNP	43.1544	2.9367	35.5388	0.9738	57.69231	1579	50	236
dl022307CNP	1.9417	2.4456	0.6897	0.8830	33.33333	1434	0	586
dp042007CNP	38.2179	2.2537		0.8951	58.59375	1027		287
dr031207CNP	32.7021	2.3349	24.4043	1.2580	42.85714	1780	29.1667	1180
ea121506CNP	61.2747	0.9682	63.0646	0.6351	82.05128	300	84.6154	186
eb031907rab69CNP		2.7494		2.1940		2324		1990
eb031907rab70CNP		2.6236		2.2609		1126		974
eb031907rab71CNP		4.7263		4.5242		1690		1588
eb032607CNP		2.7202		0.0845		1629		127
ee040507CNP		2.3827		0.0179		1053		45
er031307CNP		0.0001		0.0001		1		1
es022607CNP		4.2261		2.8253		2050		1159
ew022807CNP		1.2586		0.6948		849		534
gh021507CNP		3.0169		2.1326		1630		1279
gs031207CNPa	50.9114	1.0390		0.2375	91.66667	1319		167
gs031207CNPb		3.3503		2.1365		2406		1258
gt030907CNP		3.1169		2.8011		2631		2421
hh031207CNPa	7.5758	0.5252	7.5758	0.0002	14.28571	309	14.2857	0

hh031207CNPb	12.8099	0.6935	12.8099	0.6558	50	686	50	681
hh031207CNPc		2.0881		0.0000		1041		0
hh031207CNPd	11.3603	1.6965	11.3603	1.5520	30	798	30	755
hs012907CNP		4.5559		2.7444		2439		1435
hs022707CNP		0.0000		0.0000		0		0
im011707CNP		2.7972		2.4632		1683		1329
is021207CNP	35.5422	0.8992	47.0309	0.7895	58.49057	276	76.4706	153
jd012607CNP		1.3540		1.0829		316		269
jd032607CNP		1.1653		0.2501		543		165
jh022307CNP		0.0825		0.0000		68		0
jl021607CNP	3.8007	2.4970	3.5647	1.5668	12	484	11.3636	424
jm021207CNP2		2.8250		2.4940		1076		875
jm021207CNP	0.0000	0.0000	0.0000	0.0000	0	0	0	0
jm031207CNP		1.1624		0.2783		1300		427
jp030907rab55CNP		0.0002		0.0000		1		0
jp030907rab56CNP		1.9467		0.1565		1258		144
jp030907rab57CNP		0.8696		0.8696		554		554
jv013007CNPa		2.1115		1.3921		1160		944
ka022707CNP		1.0286		0.5537		1916		991
kc030507rab52CNP		0.0000		0.0000		0		0
kc030507rab53CNP	0.7273	0.0002	9.0909	0.0001	0	0	0	0
kg012607CNP		0.0004		0.0004		1		1
km010807CNP		0.0003		0.0003		2		2
kp012907CNP	6.6398	1.9679	25.8216	0.4433	14.81481	1258	50	261
kr030807CNP		1.5040		0.0051		539		6
kr042007CNP		1.4703		0.6466		1396		600
ks021207CNP		0.0000		0.0000		0		0
lf031907CNP		0.9609		0.2913		387		107
ll040307CNP		1.7470		0.1049		705		113
ls012607CNP	0.6329	0.3099	60.0000	0.3096	2.941176	471	100	470
lt030207CNP	2.2381	0.0004	1.4049	0.0000	0	0	0	0
md033007CNP	0.1239	0.0080	0.1239	0.0000	0	2	0	0
mj020907CNP		0.0002		0.0000		1		0
mm012607CNP	0.0587	0.0001	0.0000	0.0000	0.492611	0	0	0
mv021607CNP	0.0000	0.0010	0.0000	0.0003	0	3	0	1
ng021207CNP		2.8346		1.8763		1866		1561
pl032907CNP	27.7034	0.8623		0.1882	50	2237		605
pp022207CNP		6.2831		2.7865		2986		1322
rb021207rab24CNP	9.3676	1.2481	43.5821	0.1625	18.18182	716	80	177
rb021207rab2CNP	38.3914	3.2024	38.3914	2.0797	66.66667	1544	66.6667	999
rb031207CNP	22.9437	3.7412	31.2665	3.2535	54.09836	1912	64.8649	1705
rm031607CNP		2.3720		0.6519		1348		483
ro030207CNP		1.0399		0.7263		1595		1278
rp041307CNP		0.0407		0.0361		33		28
rr031207CNP		1.7267		1.6845		699		692

se022307CNP		2.6187		0.2180		1547		187
sf012907CNP		0.4669		0.4182		169		138
sg021207CNP	31.1884	2.6342	31.5981	2.3636	48.27586	1615	50	1465
sh040307CNP	26.2978	2.1153	26.2978	1.3880	14.28571	1093	14.2857	742
si032307CNP	4.1299	1.5643	7.6900	0.7660	12.38938	971	15.3846	589
ss021207CNP	56.5217	2.2866	20.8333	0.0128	89.47368	1041	100	56
ss032007CNP		0.0004		0.0004		2		2
st032107rab72CNP	83.8565	2.2294	83.8565	0.3189	100	1272	100	338
st032107rab73CNP		1.5046		0.2726		1290		426
sw021507CNP	10.7456	0.5088	1.9157	0.1519	42.85714	122	20	32
tb013107CNPb		0.2754		0.2555		110		104
tb022807CNP		0.0016		0.0014		7		6
ts032607CNP		2.6332		1.8201		1922		1352
wc031407CNP		0.0002		0.0000		1		0
wg030107CNP		0.0006		0.0004		3		2
wp020907CNP		1.3857		0.1872		574		143
wr030907CNP		2.3844		0.6993		3217		638
Average	23.0443	1.5952	25.0554	0.8975	41.2238	1014.5960	42.2096	583.8788

Table G3 Scoring of flash removed images for test set using Standard Deviation Method

Image	Standard Deviation Method							
	Pixel Based				Region Based			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
DSC00189		1.7874		0.8138		2161		1615
aa021907CNP		0.2896		0.2896		251		251
ab021407CNPl	59.3518	6.3034	60.3669	4.5398	78.1513	2965	80.5825	2160
ad021607CNP		0.0390		0.0372		73		64
ag021607CNP		1.9021		1.0802		962		654
ar031607rab67CNP		1.0328		0.6590		1019		606
ar031607rab68CNP		0.7603		0.7592		429		427
as030107CNP	16.8274	5.0683	15.3456	4.0703	38	7062	38.6364	5282
bj032907CNP	81.3880	4.7209	100.0000	4.3765	100	2388	100	1985
bm030607cnp	73.8875	1.3846	26.5625	0.5716	77.7778	1574	60	1182
bs031407CNP		4.5996		3.3062		2443		1669
ca022807CNP	13.1213	2.5076	13.4146	1.2535	45.4545	971	40	644
cd030907CNP2		3.4539		3.4064		1947		1908
ch041907CNP	50.1326	3.1845	50.1326	2.8032	75.0000	2505	75	2248
cm031507CNP	29.4574	2.2450	29.4342	1.6078	57.1429	3086	46.1538	2071
cs021507CNP	3.8976	1.1963	4.6698	1.0690	33.3333	1146	41.6667	945
cw021907CNP	1.2915	0.3096		0.2588	0.0000	264		219
db021207CNP	38.6256	0.2774		0.0249	75.0000	779		85
db032607CNP	20.9003	5.2040		2.6792	57.1429	3006		1306
dk031607CNP	22.4535	2.7866	2.64650	0.4002	30.7692	1857	12.5	156
dl022307CNP	1.9417	1.8712	0	1.3229	33.3333	1114	0	734
dp042007CNP	47.1776	2.8221		1.0414	64.8438	1284		286
dr031207CNP	15.6349	1.7967	19.7156	0.8666	32.8571	1769	29.1667	877
ea121506CNP	55.2239	4.3807	57.5770	1.4375	74.3590	2379	71.7949	796
eb031907rab69CNP		3.2701		2.6008		2639		2292
eb031907rab70CNP		3.5253		3.2952		1664		1551
eb031907rab71CNP		6.0478		5.9395		1955		1906
eb032607CNP		2.2773		0.0776		1882		114
ee040507CNP		1.2878		0.0136		936		39
er031307CNP		0.7558		0.7532		822		808
es022607CNP		6.0527		4.0466		2662		1669
ew022807CNP		1.4205		0.7273		982		676
gh021507CNP		4.0526		3.3839		1894		1541
gs031207CNPa	52.4799	2.6150		0.5225	91.6667	2204		445
gs031207CNPb		6.5208		4.4948		3058		1789
gt030907CNP		3.6746		3.4461		2696		2540
hh031207CNPa	10.7656	1.6755	10.7656	0.9633	14.2857	1012	14.2857	761

hh031207CNPb	15.0826	0.6813	15.0826	0.6502	50	549	50	527
hh031207CNPc		1.3827		0.3020		949		369
hh031207CNPd	12.2867	1.6283	12.2867	1.6141	30	899	30	882
hs012907CNP		5.0350		3.3152		2822		1786
hs022707CNP		0.0121		0.0104		15		10
im011707CNP		3.1993		2.3266		2252		1425
is021207CNP	54.2169	3.1696	56.5321	2.0413	86.7925	2407	94.1176	1339
jd012607CNP		1.5213		1.0715		400		337
jd032607CNP		2.5656		1.2269		1703		427
jh022307CNP		2.1014		0.1867		1360		324
jl021607CNP	20.2419	3.1587	20.3785	1.7040	32	633	34.0909	536
jm021207CNP2		3.1711		2.2758		1271		875
jm021207CNP	23.0539	5.0521	23.0539	4.8803	22.2222	1639	22.2222	1572
jm031207CNP		0.9983		0.3370		2255		496
jp030907rab55CNP		0.1104		0.1094		158		151
jp030907rab56CNP		2.2009		0.1239		2318		149
jp030907rab57CNP		0.7942		0.7941		567		566
jv013007CNPa		2.2962		1.7289		1395		1248
ka022707CNP		1.0249		0.4498		2199		962
kc030507rab52CNP		2.9494		2.3955		2155		1744
kc030507rab53CNP	5.9394	2.3081	9.0909	1.0706	12.5	2767	0	886
kg012607CNP		0.7922		0.7920		350		349
km010807CNP		2.7762		2.2096		2203		1879
kp012907CNP	8.1489	0.7843	30.0469	0.4663	11.1111	927	25	299
kr030807CNP		2.5329		0.1511		1345		98
kr042007CNP		2.0936		0.7265		1919		758
ks021207CNP		0.9067		0.9067		738		738
lf031907CNP		3.3617		1.1450		1973		440
ll040307CNP		2.4632		0.1415		933		143
ls012607CNP	2.2785	0.2638	40	0.2633	11.7647	325	100	323
lt030207CNP	6.0476	0.3673	9.8340	0.3667	20	311	40	311
md033007CNP	35.1508	0.1934	35.1508	0.1843	90	175	90	173
mj020907CNP		0.0021		0.0019		3		2
mm012607CNP	0.05866	0.3115	0	0.3109	0.4926	422	0	421
mv021607CNP	6.93993	0.9165	12.7666	0.6145	12.2222	455	20	289
ng021207CNP		3.3045		2.6392		2277		1861
pl032907CNP	25.8496	1.6578		0.2098	75	2795		574
pp022207CNP		5.1374		2.2020		2408		1092
rb021207rab24CNP	7.2595	1.1231	4.7761	0.3087	9.0909	737	20	275
rb021207rab2CNP	21.6418	2.1493	21.6418	1.2730	50.0000	1492	50	1004
rb031207CNP	37.2872	4.5286	49.0106	3.6279	73.7705	2276	86.4865	1975
rm031607CNP		2.2855		1.2144		1362		734
ro030207CNP		0.9388		0.8627		1749		1572
rp041307CNP		2.7679		2.7510		846		837
rr031207CNP		1.4392		1.3272		577		569

se022307CNP		1.0485		0.1755		859		175
sf012907CNP		0.4211		0.3866		175		144
sg021207CNP	38.5507	3.0796	39.2857	2.5265	79.3103	1766	84.6154	1469
sh040307CNP	26.9823	1.5624	26.9823	1.1445	23.8095	1752	23.8095	1233
si032307CNP	8.3160	2.3568	16.7649	1.1795	20.3540	1379	25.6410	751
ss021207CNP	47.7226	1.5471	16.6667	0.0081	84.2105	1153	100	31
ss032007CNP		0.2109		0.2109		230		230
st032107rab72CNP	0	1.8456	0	0.3618	0	1356	0	469
st032107rab73CNP		1.1497		0.2523		1259		488
sw021507CNP	41.4474	1.6903	49.8084	0.4868	100	544	100	190
tb013107CNPb		3.9773		3.6089		1682		1527
tb022807CNP		1.9511		0.7149		5639		2359
ts032607CNP		3.0825		2.5291		2280		1760
wc031407CNP		1.4147		1.1500		895		775
wg030107CNP		1.7897		1.6500		1356		1254
wp020907CNP		2.5919		0.1646		899		91
wr030907CNP		3.0627		1.0247		3129		781
Average	25.9765	2.2661	25.8762	1.4124	46.8442	1550.5455	47.2285	922.7778

Table G4 Scoring of flash removed images for test set using Standard Deviation Method combined with salient feature masks

Image	Standard Deviation Method combined with Salient Point							
	Pixel Based				Region Based			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
DSC00189		1.7883		0.8146		2183		1637
aa021907CNP		0.2897		0.2897		259		259
ab021407CNPI	59.8654	6.3094	60.9390	4.5451	78.15126	2957	80.5825	2150
ad021607CNP		0.0390		0.0372		75		65
ag021607CNP		1.9104		1.0829		1007		680
ar031607rab67CNP		1.0345		0.6601		1032		611
ar031607rab68CNP		0.7605		0.7593		434		432
as030107CNP	16.8623	5.0689	15.3836	4.0704	38	7068	38.6364	5283
bj032907CNP	82.3344	4.7247	100.0000	4.3783	100	2396	100	1989
bm030607cnp	74.3073	1.3875	28.5156	0.5730	77.77778	1589	60	1184
bs031407CNP		4.6074		3.3064		2471		1675
ca022807CNP	13.1213	2.5094	13.4146	1.2540	45.45455	984	40	654
cd030907CNP2		3.4614		3.4139		1991		1950
ch041907CNP	50.1326	3.1873	50.1326	2.8036	75	2518	75	2248
cm031507CNP	29.4574	2.2466	29.4342	1.6083	57.14286	3090	46.1538	2071
cs021507CNP	4.7327	1.2025	5.6704	1.0714	40	1220	50	997
cw021907CNP	1.2915	0.3105		0.2588	0	268		219
db021207CNP	38.6256	0.2823		0.0254	75	819		86
db032607CNP	20.9003	5.2107		2.6804	57.14286	3056		1315
dk031607CNP	22.4535	2.7921	2.6465	0.4002	30.76923	1884	12.5	156
dl022307CNP	1.9417	1.8719	0	1.3229	33.33333	1120	0	735
dp042007CNP	47.9269	2.8575		1.0425	68.75	1433		305
dr031207CNP	15.7077	1.8011	19.7156	0.8683	32.85714	1801	29.1667	894
ea121506CNP	55.2642	4.3879	57.6192	1.4400	74.35897	2406	71.7949	799
eb031907rab69CNP		3.2970		2.6251		3053		2674
eb031907rab70CNP		3.5288		3.2979		1729		1611
eb031907rab71CNP		6.0541		5.9458		2030		1980
eb032607CNP		2.2783		0.0776		1895		114
ee040507CNP		1.2884		0.0141		936		39
er031307CNP		0.7573		0.7543		834		816
es022607CNP		6.0565		4.0466		2676		1671
ew022807CNP		1.4229		0.7273		996		676
gh021507CNP		4.0560		3.3845		1916		1547
gs031207CNPa	52.4799	2.6210		0.5235	91.66667	2222		445
gs031207CNPb		6.5212		4.4950		3062		1789
gt030907CNP		3.6879		3.4526		2744		2575
hh031207CNPa	10.7656	1.6792	10.7656	0.9640	14.28571	1025	14.2857	768

hh031207CNPb	15.0826	0.6843	15.0826	0.6527	50	571	50	543
hh031207CNPc		1.3885		0.3078		985		405
hh031207CNPd	12.2867	1.6292	12.2867	1.6149	30	900	30	882
hs012907CNP		5.0356		3.3157		2833		1794
hs022707CNP		0.0121		0.0104		15		10
im011707CNP		3.2034		2.3266		2288		1427
is021207CNP	55.2209	3.1818	56.7696	2.0424	86.79245	2440	94.1176	1340
jd012607CNP		1.5251		1.0741		411		346
jd032607CNP		2.5664		1.2269		1704		427
jh022307CNP		2.1058		0.1867		1379		324
jl021607CNP	21.4512	3.3858	21.7035	1.7813	36	1065	38.6364	691
jm021207CNP2		3.1725		2.2772		1289		892
jm021207CNP	25.4491	5.0622	25.4491	4.8870	33.33333	1671	33.3333	1591
jm031207CNP		0.9987		0.3370		2265		496
jp030907rab55CNP		0.1109		0.1095		160		152
jp030907rab56CNP		2.2009		0.1239		2318		149
jp030907rab57CNP		0.7945		0.7945		580		579
jv013007CNPa		2.3072		1.7311		1428		1262
ka022707CNP		1.0315		0.4499		2237		962
kc030507rab52CNP		2.9497		2.3955		2165		1754
kc030507rab53CNP	5.9394	2.3116	9.0909	1.0710	12.5	2791	0	888
kg012607CNP		0.7922		0.7920		363		362
km010807CNP		2.7777		2.2098		2237		1904
kp012907CNP	8.1489	0.7843	30.0469	0.4663	11.11111	929	25	301
kr030807CNP		2.5345		0.1528		1366		119
kr042007CNP		2.0946		0.7265		1928		758
ks021207CNP		0.9069		0.9069		738		738
lf031907CNP		3.3639		1.1450		1995		443
ll040307CNP		2.4657		0.1415		1028		143
ls012607CNP	2.2785	0.2638	40	0.2633	11.76471	327	100	325
lt030207CNP	6.0476	0.3680	9.8340	0.3674	20	314	40	314
md033007CNP	35.8530	0.2009	35.8530	0.1913	90	182	90	179
mj020907CNP		0.0021		0.0019		3		2
mm012607CNP	0.0733	0.3124	0.1217	0.3118	0.985222	436	2.5641	435
mv021607CNP	7.0530	0.9222	13.0104	0.6174	12.22222	484	20	301
ng021207CNP		3.3065		2.6407		2315		1891
pl032907CNP	26.4676	1.6584		0.2098	75	2801		574
pp022207CNP		5.1412		2.2021		2418		1091
rb021207rab24CNP	7.2595	1.1235	4.7761	0.3091	9.090909	747	20	285
rb021207rab2CNP	21.8076	2.1502	21.8076	1.2736	50	1496	50	1006
rb031207CNP	37.3449	4.5683	49.1425	3.6650	75.40984	2990	89.1892	2658
rm031607CNP		2.2869		1.2152		1371		745
ro030207CNP		0.9396		0.8635		1753		1576
rp041307CNP		2.8371		2.8202		990		981
rr031207CNP		1.4446		1.3286		606		591

se022307CNP		1.0485		0.1755		859		175
sf012907CNP		0.4218		0.3866		180		144
sg021207CNP	38.8986	3.0862	39.6489	2.5288	79.31034	1806	84.6154	1492
sh040307CNP	27.2675	1.5651	27.2675	1.1470	33.33333	1805	33.3333	1283
si032307CNP	8.6367	2.3723	17.2068	1.1858	24.77876	1465	28.2051	768
ss021207CNP	47.7226	1.5476	16.6667	0.0081	84.21053	1160	100	32
ss032007CNP		0.2128		0.2128		235		235
st032107rab72CNP	0	1.8468	0	0.3618	0	1361	0	469
st032107rab73CNP		1.1572		0.2523		1302		488
sw021507CNP	41.4474	1.6908	49.8084	0.4872	100	545	100	191
tb013107CNPb		4.0034		3.6332		2038		1856
tb022807CNP		1.9593		0.7230		5668		2382
ts032607CNP		3.0825		2.5291		2281		1760
wc031407CNP		1.4178		1.1509		941		793
wg030107CNP		1.7957		1.6501		1371		1258
wp020907CNP		2.5921		0.1646		901		91
wr030907CNP		3.0627		1.0247		3129		781
Average	26.2477	2.2733	26.1709	1.4159	47.8883	1591.2929	48.4445	948.8182

Table G5 Scoring of flash removed images for test set using Curve fitting Method

Image	Curve Fitting Method							
	Pixel Based				Region Based			
	Whole Image		Lesion Area		Whole Image		Lesion Area	
	True Positive	False Positive	True Positive	False Positive	True Positive	False pos Detection	True Positive	False pos Detection
DSC00189		2.1696		0.1285		1008		100
aa021907CNP		1.8614		0.1445		2079		166
ab021407CNPl	39.0542	1.5692	41.4677	1.1318	63.8655	535	66.0194	365
ad021607CNP		0.0080		0.0080		8		8
ag021607CNP		0.0403		0.0355		34		30
ar031607rab67CNP		0.5650		0.1248		541		147
ar031607rab68CNP		0.5751		0.5738		418		416
as030107CNP	39.5468	2.6715	42.1687	2.6677	58.0000	735	63.6364	728
bj032907CNP	21.4511	0.7763	100.0000	0.6838	50.0000	639	100.0000	568
bm030607cnp	42.4013	1.0673	17.5781	0.1472	70.3704	1904	40.0000	374
bs031407CNP		2.1470		0.8682		748		278
ca022807CNP	0.0000	0.0728	0.0000	0.0004	0.0000	33	0.0000	1
cd030907CNP2		0.6700		0.6670		161		149
ch041907CNP	47.8780	2.8821	47.8780	2.4016	87.5000	1353	87.5000	1186
cm031507CNP	7.6966	0.0325	8.8366	0.0119	21.4286	49	23.0769	25
cs021507CNP	28.0067	1.3586	23.8159	0.2454	36.6667	1828	25.0000	505
cw021907CNP	0.2026	0.0274		0.0272	0.0000	25		25
db021207CNP	18.8389	0.0443		0.0329	75.0000	97		93
db032607CNP	15.7556	1.4987		0.6105	42.8571	1028		403
dk031607CNP	38.8828	2.3632	35.5388	0.9836	50.0000	878	50.0000	235
dl022307CNP	1.9417	2.1136	0.6897	0.2898	33.3333	1209	0.0000	153
dp042007CNP	22.9132	0.0986		0.0069	40.6250	22		6
dr031207CNP	30.5899	1.1237	23.4435	0.2073	42.8571	486	27.0833	149
ea121506CNP	61.9605	3.9873	63.7822	1.0813	84.6154	1505	87.1795	505
eb031907rab69CNP		0.0566		0.0090		184		24
eb031907rab70CNP		0.6511		0.4396		260		199
eb031907rab71CNP		2.6827		2.1582		670		489
eb032607CNP		2.2425		0.0444		842		42
ee040507CNP		1.7951		0.0003		905		2
er031307CNP		2.5885		0.4335		2002		398
es022607CNP		2.0576		1.3319		1745		782
ew022807CNP		1.0676		0.5040		591		279
gh021507CNP		1.2602		0.4033		607		300
gs031207CNPa	40.8224	0.3609		0.1720	75.0000	72		23
gs031207CNPb		0.9663		0.7242		1064		467
gt030907CNP		1.2950		1.0789		1201		1042
hh031207CNPa	12.0415	1.1100	12.0415	0.9479	42.8571	704	42.8571	582

hh031207CNPb	1.2397	0.4168	1.2397	0.4166	0.0000	287	0.0000	286
hh031207CNPc		0.8673		0.2102		561		205
hh031207CNPd	9.3613	0.5999	9.3613	0.4728	25.0000	197	25.0000	174
hs012907CNP		3.1193		2.2055		1288		852
hs022707CNP		0.0175		0.0167		17		15
im011707CNP		0.0070		0.0031		10		7
is021207CNP	45.4485	2.4687	49.6437	1.5642	75.4717	1760	82.3529	893
jd012607CNP		0.2195		0.2013		48		43
jd032607CNP		2.2456		0.5413		1229		365
jh022307CNP		0.1156		0.0331		115		47
jl021607CNP	0.8926	0.2166	0.4416	0.1766	2.0000	49	0.0000	43
jm021207CNP2		0.7059		0.5351		226		129
jm021207CNP	0.0000	0.0000	0.0000	0.0000	0.0000	0	0.0000	0
jm031207CNP		1.1237		0.1320		763		101
jp030907rab55CNP		0.1038		0.0432		103		72
jp030907rab56CNP		0.0006		0.0000		2		0
jp030907rab57CNP		0.0002		0.0002		1		1
jv013007CNPa		0.7929		0.3324		322		119
ka022707CNP		0.0030		0.0011		17		5
kc030507rab52CNP		1.6391		1.2846		1067		877
kc030507rab53CNP	27.2727	1.6446	68.1818	0.3582	31.2500	1843	100.0000	426
kg012607CNP		0.6339		0.6283		247		235
km010807CNP		0.3048		0.2582		224		172
kp012907CNP	14.6881	2.9836	25.8216	0.4027	25.9259	1742	50.0000	167
kr030807CNP		1.9115		0.2148		1221		144
kr042007CNP		0.2948		0.0000		65		0
ks021207CNP		0.0676		0.0597		94		82
lf031907CNP		1.2338		0.3242		785		167
ll040307CNP		1.6720		0.0605		634		62
ls012607CNP	0.0000	0.8044	0.0000	0.8041	0.0000	1135	0.0000	1134
lt030207CNP	3.5238	0.1083	6.6411	0.1065	20.0000	136	40.0000	130
md033007CNP	34.4899	0.1335	34.4899	0.1329	90.0000	124	90.0000	122
mj020907CNP		0.0002		0.0000		1		0
mm012607CNP	0.9531	0.0334	0.2435	0.0205	0.9852	142	0.0000	35
mv021607CNP	4.7774	0.4346	10.2072	0.1888	10.0000	421	20.0000	128
ng021207CNP		2.0816		1.1447		1018		736
pl032907CNP	0.8239	0.0414		0.0011	0.0000	51		2
pp022207CNP		4.4214		1.8740		1594		602
rb021207rab24CNP	12.3791	1.6790	43.5821	0.0946	21.8182	721	80.0000	58
rb021207rab2CNP	5.4726	2.2243	5.4726	1.1018	12.5000	916	12.5000	372
rb031207CNP	19.4228	0.9007	27.2427	0.5802	47.5410	289	56.7568	123
rm031607CNP		1.5835		0.3169		698		243
ro030207CNP		0.0437		0.0181		191		43
rp041307CNP		2.0755		2.0739		566		563
rr031207CNP		0.0021		0.0006		4		1

se022307CNP		1.8154		0.0004		1244		2
sf012907CNP		0.0165		0.0165		6		6
sg021207CNP	32.0580	2.1611	32.5061	1.9912	44.8276	1017	46.1538	960
sh040307CNP	0.0000	1.1390	0.0000	0.6636	0.0000	609	0.0000	385
si032307CNP	1.2350	0.0083	1.4437	0.0013	4.4248	11	2.5641	2
ss021207CNP	52.4845	1.2724	0.0000	0.0042	84.2105	601	0.0000	8
ss032007CNP		0.0226		0.0226		51		51
st032107rab72CNP	83.8565	2.0693	83.8565	0.1588	100.0000	1060	100.0000	126
st032107rab73CNP		1.3168		0.1090		963		133
sw021507CNP	14.2544	0.7767	8.4291	0.2226	71.4286	306	60.0000	108
tb013107CNPb		2.8666		2.6811		1191		1089
tb022807CNP		1.4900		0.5952		3499		1635
ts032607CNP		0.1836		0.0171		213		33
wc031407CNP		1.3903		0.6091		633		371
wg030107CNP		1.6881		0.8334		1535		687
wp020907CNP		0.8871		0.0053		416		11
wr030907CNP		1.3739		0.5480		1731		448
Average	20.8654	1.1143	24.2954	0.4923	38.5590	668.4848	40.5200	275.5556

BIBLIOGRAPHY

[1] Jemal A., Siegel R., Ward E., Murray T., Xu J., and Thun M.J. (2007) "Cancer Statistics 2007," CA: *A Cancer Journal for Clinicians*, 57(1): 43-66.

[2] X. Chen. Automatic Detection of Lesion Border and Edge-related Structures in Dermoscopy Images, PhD Dissertation in Electrical Engineering, University of Missouri-Rolla, 2007.

[3] T. Lee, V. Ng, R. Gallagher, A. Coldman, D. McLean, "DullRazor: a Software Approach to Hair Removal", *Comp Biol Med,* Vol 27(6): pp. 533-43, 1997.

[4] X. Chen, R.H. Moss, W.V. Stoecker, T.K. Lee, R.J. Stanley, B. Shrestha, D.I. McLean. Software Improvements in Hair Detection using DullRazor. Proceedings of the 6th World Congress on Melanoma. Vancouver, Canada. Sep. 6-10, 2005 (2005).

[5] Robert Paul McLean, "Tumor Classification Based on Relative Color Analysis of Melanoma and Non-Melanoma Tumor Images," Master Thesis, University of Missouri-Rolla, 1994.

[6] S. Umbaugh, "Computer Vision in Medicine: Color Metrics and Image Segmentation Methods for Skin Cancer Diagnosis," Doctor of Philosophy in Electrical Engineering Dissertation, University of Missouri - Rolla, 1990.

[7] Faziloglu Y, Stanley RJ, Moss RH, Stoecker WV, McLean RP. Color histogram analysis for melanoma discrimination in clinical images. *Skin Research and Technology* 2003:9:147-155.

[8] Chen J, Stanley R, Moss RH, Stoecker WV. Color analysis of skin lesion regions for melanoma discrimination in clinical images. *Skin Research and Technology* 2003: 9:94-104.

[9] Stanley RJ, Moss RH, Stoecker WV, Aggarwal C. A fuzzy-based histogram analysis technique for skin lesion discrimination in dermatology clinical images. *Computerized Medical Imaging and Graphics* 2003: 27:387-396.

[10] Stanley RJ, Stoecker WV, Moss RH. A relative color approach to color discrimination for malignant melanoma detection in dermoscopy images. *Skin Research and Technology* 2007:13(1):62-72.

[11] Z. Zhang, "Automatic Skin Tumor Segmentation and Diagnosis." Doctor of Philosophy in Electrical Engineering Dissertation, University of Missouri - Rolla, 1997.

[12] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66.

[13] P Jella. Pigment network extraction and salient point analysis, Master Thesis in Electrical and Computer Engineering Department, University of Missouri-Rolla, 2004.

[14] S. Umamaheswaran. Feature and color analysis of images of the cervix. Master Thesis in Electrical and Computer Engineering Department, University of Missouri-Rolla, 2006.

[15] Sobel IE. Camera Models and Machine Perception, Ph.D. Thesis, Electrical Engineering Department, Stanford University, Stanford, CA, 1970.

VITA

Radhika Nagane was born in Solapur, India, on March 1, 1983. She received her primary and secondary education in Solapur and graduated from Walchand College of Arts and Science, Solapur in 2000. She graduated with a Bachelor of Engineering degree in Electronics and Telecommunication Engineering from the Walchand Institute of Technology, Shivaji University, Kolhapur, India in May 2004. She later worked as a software Engineer at National Stock Exchange IT, Mumbai, India.

She enrolled as graduate student in the University of Missouri-Rolla in August 2005, majoring in Electrical Engineering. She was a Graduate Research Assistant in the DERMVIS group of cancer research from January 2006 to May 2007 and received her MS degree in Electrical Engineering in December 2007.