Development of Automated, Interactive Software for Skin Cancer Diagnosis using Image Analysis and Multivariate Modeling Techniques

February 5, 2016

Abstract

Skin cancer, the most common form of cancer in the United States, can be fatal if not carefully monitored and treated. Some forms of skin cancer, such as melanoma, can be mistaken as benign and given time to develop and expand. However, if the patient is diagnosed early on before the cancer develops, full recovery is possible. An accessible skin lesion monitoring software could allow patients to check skin lesions more frequently. Multiple prior skin cancer diagnosis programs are inaccurate, utilize only traditional variables and algorithms, or lack complete automation capabilities. This project aimed to develop an interactive, automated software for accurately identifying cancerous growths. First, JMP Scripting Language algorithms were created to segment lesion outer borders from pictures. Then, 44 characteristic metrics, such as texture, symmetry, border irregularity, and variation of color, were extracted from images. Multiple novel, significant variables and algorithms were introduced. A predictive discriminant analysis model was built from these collected metrics, achieving a misclassification rate of 5.2% from 152 samples. A user interface was programmed to allow users to analyze their own images. Overall, my software can efficiently and accurately analyze skin lesion malignancy, which can aid in early and reliable identification of skin cancer.

1. Introduction: Purpose, Related Work, and Implications

Skin cancer, the most common form of cancer in the United States, can be fatal if not carefully monitored and treated. During regular checkups, some forms of skin cancer, such as melanoma, can be mistaken for benign growths and given time to develop and expand. Detection of suspicious lesions early on is key, as there are historically good odds for full recovery with swift attention and treatment [19]. Since the visit to a doctor's office is often inaccessible or cumbersome for many people, preventing consistent screening, a skin lesion monitoring software would provide an additional, more accessible screening stage, allowing patients to check their skin lesions more frequently. The objective of this study was to develop an interactive, automated software for separating benign moles from cancerous growths, and to find pertinent variables that contributed to a more accurate diagnosis.

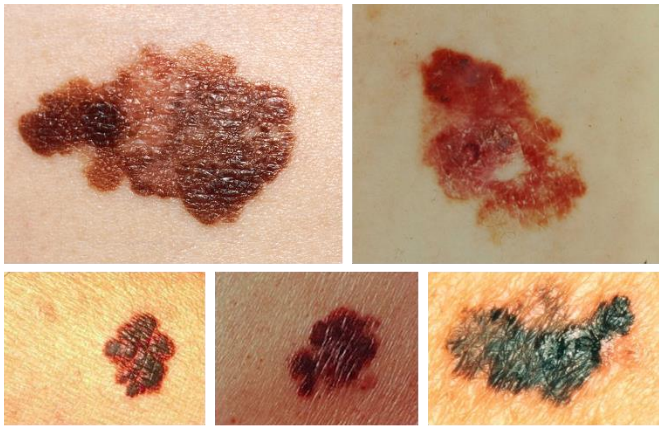


Figure 1. Examples of cutaneous malignant melanoma (Images from [1], [11], [12], [13])

There have been multiple previous skin cancer diagnosis programs and models, some made into mobile applications, which have had a few drawbacks: being inaccurate, analyzing only traditional variables, or lacking complete automation capabilities.

One previous investigation compared Otsu's method, K-mean Clustering, and Gradient Vector Flow, finding Otsu's method most effective [5], and two other studies conducted with images taken on mobile devices confirmed Otsu's method's efficacy [17, 18]. Another researcher used Gaussian smoothing processes to identify border irregularity [8]. Other researchers analyzed traditional variables using dermoscopy images [14]. Mobile apps have also used fractal geometry for segmentation [20]. However, most of this past research has required access to expensive software that have specialized libraries and functions, such as MatLab.

In this skin lesion analysis software, I made improvements on all three drawbacks by generating new metrics and algorithms to improve accuracy, introducing texture, a novel and effective analysis variable not used in previous texture-based studies (see discussion for details) [3], developing a new rotational symmetry analysis algorithm [5], and completely automating analysis, which allows for image batch processing. To simplify and make algorithms library-independent, I programmed the entire software in JMP Scripting Language (JSL), a language similar to javascript that can be translated for implementation in other programming languages.

My algorithm analyzes lesion borders using edge detection rather than 'knock-out' thresholding (isolating the lesion as an area [17]) to preserve image data in the lesion's interior. This choice in technique allowed my program to incorporate a novel (not mentioned in related literature [3]) edge-based texture variable, which was the most significant variable in the model.

This software is also completely automated and has high accuracy, allows the user to see every step of the analysis, visualizes the diagnosis relative to other samples on a plot, and communicates a classification confidence measure. The software is intended for usage in screening by both patients and physicians.

My software can also be translated into other highly accessible programming languages, such as R, Python, and Java, allowing it to be more easily improved during future development.

2. Evaluation Variables, Data Set, and Materials

2.1 Traditional and Novel Evaluation Variables

The five traditional criteria for evaluating a skin lesion are known as the ABCDE rule: asymmetry, border irregularity (scalloped or blurry edges), color variation, diameter (size), and evolution (changes over time) [1].

However, software cannot analyze diameter without scale, which would burden the user to place an object of reference next to their lesion at a non-distortive angle. Evolution also cannot be evaluated without guaranteeing multiple images of the same lesion being taken from the same angle and distance. Therefore, in this study, the first three of the ABCDE criteria are assessed. A fourth variable, texture, was also added, which measures the variation of appearance in the interior of the lesion (whether color, roughness, or shading) through analyzing the amount of extraneous edges detected by the canny edge detection filter. This was later found to be the most significant variable in the model. For these four primary variables, 44 metrics were produced.

2.2 Data Set and Materials

Image samples, 152 in total, were collected via online search for both benign and malignant samples. The sample images were taken in good focus, as close as possible to the lesion with a digital camera. As needed, images were adjusted for overexposure and brightness. Distracting features, such as the outline of an arm, were cropped out. For this study, two classes of lesions, malignant and benign, were each represented by 76 samples.

JSL, the programming language of JMP software, was the programming language used throughout this project. Though similar to javascript, JSL is especially designed for data table manipulation. This project leveraged this capability, creating a new use (not extensively explored before in JMP, only basic manipulation [16]) for the conversion of pixel data to data tables.

3. Methodology

3.1 Image Segmentation Algorithm

First, a clear outer border of the original lesion was segmented. The resulting automated JSL program processed images in five steps: conversion to grayscale, filter application and edge detection, noise removal, outer border detection, and cropping for fit.

3.1.1 Conversion to Grayscale

The conversion of a color image to grayscale combines RGB values in a ratio of 30% red, 59% green, and 11% blue [4]. Conversion to grayscale allows for better feature extraction, since the image usually will have fewer distracting features due to only containing pixel intensity data.

3.1.2 Filter Application and Edge Detection

Some standard image filters are applied, supported at the operating system level. The three filters used in this study were `despeckle`, `reduce noise`, and `canny`, which respectively "removes defects" from images, reduces "random variation", and creates a binary mask of the image by detecting any edges it finds (clear boundaries between pixels, usually differentiation by color) and outlining them in white [6]. The `reduce noise` filter has a numeric parameter that controls the amount of smoothing applied to the image.

```
imgEdge = newImage(imgBW);
imgEdge << filter("despeckle");
imgEdge << filter("reduce noise", 2.2);
imgEdge << filter("canny");
```

After application of these three filters, all detectable edges in the image are displayed in white.

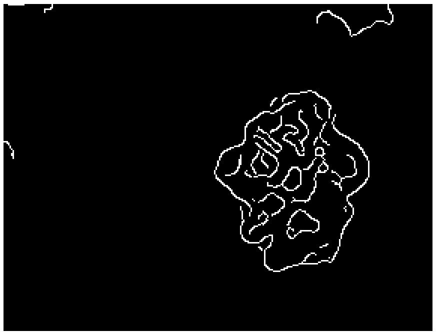


Figure 2. Step two in image segmentation is detection of all edges in the image

3.1.3 Noise Removal

Multiple methods were considered for the noise removal algorithm, including image smoothing before border detection, which would decrease the quality of the image. To avoid data loss, the final algorithm took into account noise on all four sides of the image that was detached from the lesion itself, which applied to the majority of noise encountered in the samples.

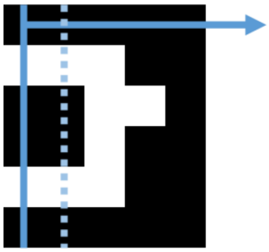


Figure 3. Noise removal algorithm illustration

The program searches in four directions for columns or rows that contain detached white segments (as seen above, if white pixels are detected in a column, but a subsequent column has none). It then compares the number of white pixels on the left and right of said current column, erasing the smaller of the two white segments (converting all pixels to black).

A snippet of the code (formatted for efficient use of space) for searching the columns from left to right is shown below. `coord[]` represents the matrix that contains the binary pixel data of the image produced from edge detection, and `numRows` and `numCols` represent the number of rows and columns in that matrix.

```
{r, g, b} = imgEdge << getPixels("rgb"); //store all RGB data for this image
...
whiteCols = 0; //number of columns that have white pixels (not completely black)
for (i = 1, i <= numCols, i++, //go through all the cols
    whtCount = 0; //number of white pixels in one col
    for(j = 1, j <= numRows, j++,
        if(coord[j, i] == 1, whtCount++;); //found the first white col
    );
    if(whtCount > 0, whiteCols++;); //if there were white pixels in this column
    if(whtCount == 0, //if you had no white pixels in this col, ++ the black
        if(whiteCols > 0, // end of continuous white piece, check which white segment is larger
            numWhiteSF = 0; //number of white pixels on "so far” side
            numWhiteTD = 0; //number of white pixels on "to do” side
            for(k = 1, k <= i, k++, //this loop sums up how much white in "so far"
                for(l = 1, l <= numRows, l++,
                    if(coord[l, k] == 1, numWhiteSF++;);
                );
            );
            for(k = i, k <= numCols, k++, //this loop sums up how much white in "to do"
                for(l = 1, l <= numRows, l++,
                    if(coord[l, k] == 1, numWhiteTD++;);
                );
            );
            if(numWhiteTD > numWhiteSF, //if “to do” white pixel amount is greater
                for(k = 1, k <= i, k++,
                    for(l = 1, l <= numRows, l++, coord[l, k] = 0;);
                );
            );
            if(numWhiteSF > numWhiteTD, //if “so far” white pixel amount is greater
                for(k = i, k <= numCols, k++,
                    for(l = 1, l <= numRows, l++, coord[l, k] = 0;);
                );
            );
            whiteCols = 0;
        );
    );
);
```

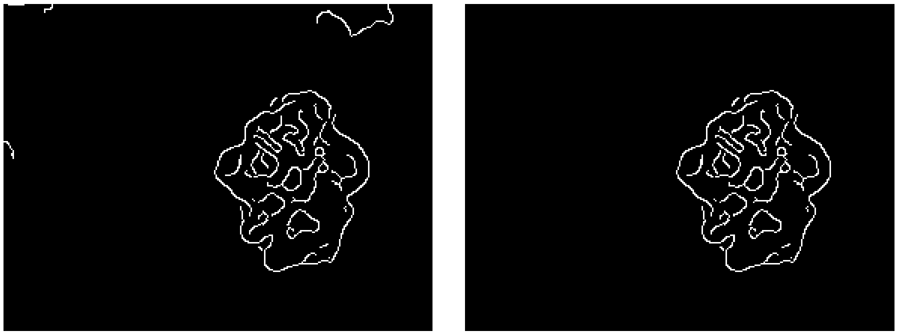


Figure 4. Step three in image segmentation is noise removal

3.1.4 Outer Border Detection

The method for detecting the lesion’s outer border was searching for the first white pixel going outward-inward, starting from the edges of the image in each direction. It was determined that searching from four directions would be sufficient enough to generate the border (rather than from all 360 degrees). The code below occurs after the noise removal. This snippet corresponds to searching for the first white pixel from the top to bottom of the image. It adds each pair of coordinates for a white pixel to `coordRow[]` and `coordCol[]`.

```
//find all the outermost white pixels on top side of the image
whiteCount = 0;
for(j = 1, j <= numCols, j++,
    whiteCount = 0;
    for(i = 1, AND(whiteCount == 0, i <= numRows), i++,
        if(coord[i, j] == 1, //if the color is white
            whiteCount++;
            insertInto(coordRow, i); //insert the white coordinate into the matrix
            insertInto(coordCol, j);
        );
    );
);
```

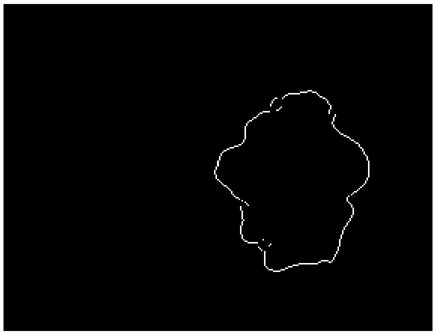


Figure 5. Fourth step in image segmentation is outer border detection

3.1.5 Cropping for Fit

This algorithm produces a cropped image of the lesion and border images with the largest dimensions that still contains the entire lesion. The cropping creates consistency among all images before variable analysis. The program goes through the lists of white pixel coordinates, searching for the leftmost (`le`), rightmost (`ri`), highest (`to`), and lowest (`bo`) white points.

```
onlyMole = newImage(outerBorder); //crop with 1 pixel of tolerance since bounds are exclusive
onlyMole << crop( left(le - 1), top(to - 1), bottom(bo + 1), right(ri + 1));
```

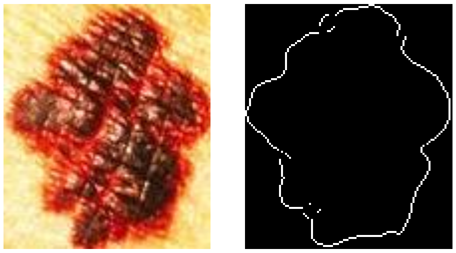


Figure 6. Original lesion image and border image cropped to fit for consistency

3.2 Variable Analysis Algorithm

Algorithms were designed and developed to analyze four primary variables: symmetry, border irregularity, color variation, and texture. Both symmetry and color variation calculations extracted multiple metrics, bringing the total to 44 total extracted metrics.

3.2.1 Symmetry Calculation

The symmetry calculation requires rotation of the image’s white pixels. To ensure rotated coordinates won’t translate outside of the image bounds, a new square image is created with side length equal to the diagonal of the cropped image. The lesion border is centered in the square.

Twelve subsequent image rotations occur, 30 degrees counterclockwise per. Before each rotation, the white pixels are transformed to their new coordinates. At each angle, the program goes down the rows that contain white pixels, comparing the two distances from the center of the image to the leftmost and rightmost pixel on each row. Each rotation yields one symmetry metric: the ratio between the number of incongruent distances (those that are considered different enough; the program’s standard is that the difference is > 5% the sum of the distances, except those <= 3 pixels in difference due to the small size of some lesions) and total pairs compared (subtracting each measurement from 1 so that symmetry lies on a scale from 0, asymmetrical, to 1, symmetrical). These are sorted ascending, and their average is the overall symmetry metric.

In the code snippet below, the ‘x’ shift and ‘y’ shift equation is the pair of equations for standard Cartesian coordinate rotation about the origin [9]. The program shifts the rotated points so that the rotation is about the center of the image and not the origin (upper-left-hand corner).

```
while(currentAngl <= endDegree,
    coordColRot = {}; coordRowRot = {}; //lists for storing the rotated coordinate values

    //rotating col & row values, using the 'x' shift equation for Col, 'y' shift eq for Row
    for(j = 1, j <= N Items(coordColShft), j++, //coordColShft contains centered image pixels
        insertInto(coordColRot, coordColShft[j] - midX); //each point is individually rotated
        insertInto(coordRowRot, coordRowShft[j] - midY);
        coordColTemp = coordColRot[j];
        coordColRot[j] = Round((coordColRot[j] * cos(currentAngl)) + (coordRowRot[j] * sin(currentAngl)));
        coordRowRot[j] = Round((coordRowRot[j] * cos(currentAngl)) - (coordColTemp * sin(currentAngl)));
        coordColRot[j] += midX;
        coordRowRot[j] += midY;
    );

    ...[here comparing distances, determining which are incongruent]

    currentAngl += increment;
);
```

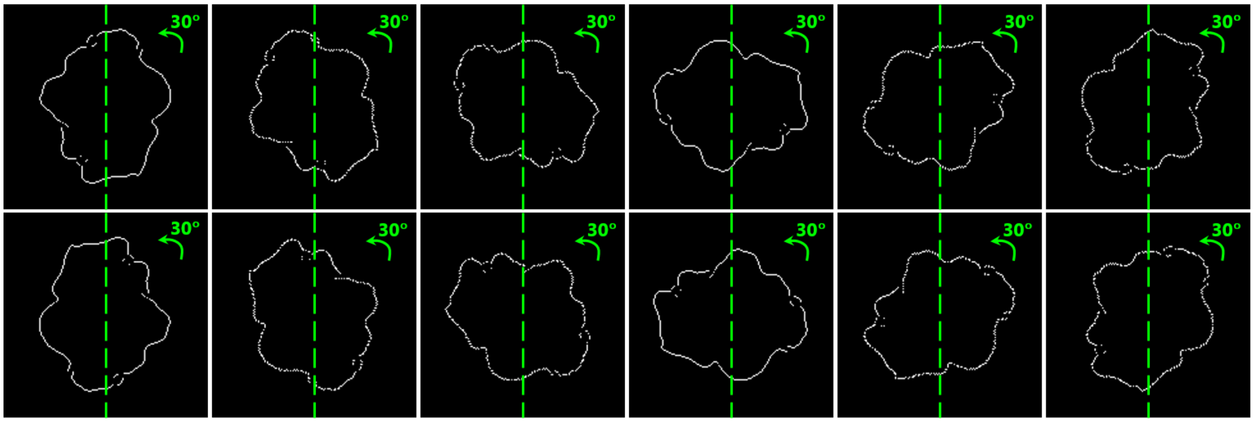


Figure 7. Symmetry analysis of malignant melanoma's outer border. Measurements taken from 30-degree rotations collectively produce 13 metrics

The coordinate point transformation equation in two dimensions used above:

$$x' = x\cos\theta + y\sin\theta$$
$$y' = -x\sin\theta + y\cos\theta \text{ [9]}$$

3.2.2 Border Irregularity Calculation

To calculate border irregularity, the slopes of each adjacent pair of border pixels are calculated. The program counts the number of times the slope of the outline changes signs (either consecutively or over multiple points; i.e. both {-1, 1} and {-1, 0, -1, 0, 2} are counted); this indicates a scallop-shaped edge, which suggests irregularity. A perfect circle has a score of 4, and the upper limit for most lesions is around 400.

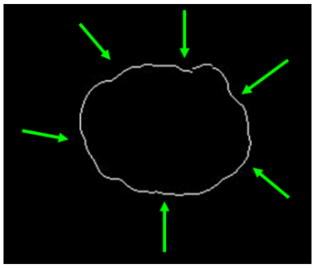


Figure 8. Border irregularity analysis of benign lesion outer border with few irregularities. Green arrows indicate examples for points of slope sign change that suggest irregularity

3.2.3 Color Variation Calculation

The color variation calculation produces the greatest number of metrics (28). Color is analyzed in two color models: RGB (red, green, blue) and HLS (hue, lightness, saturation) [4].

Intensity, the lightness or darkness of an image in grayscale, is also calculated. The number of significant r, g, b, h, l, s and i values, the maximum and minimum values and their respective bins' indices are then all calculated. Bins, or ranges, are created for each of the seven color components (in this study, 100 bins were used). A bin's range of values is considered significant if the number of pixels in that variable's bin ≥ the numerical average of pixel counts across all the bins for that variable. Finally, the sum of the significant bins for all seven variables is considered the overall color variation metric. Each count metric lies on a scale from 1 to a multiple of 100, and index metrics lie on a scale from 1 to 100.

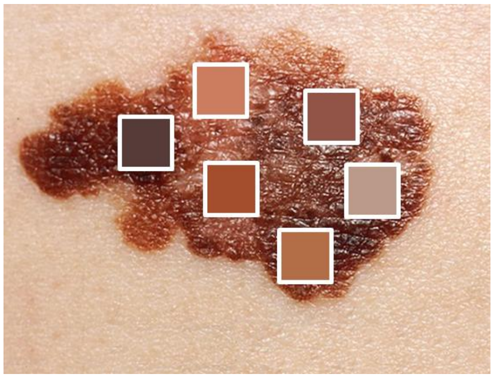


Figure 9. Color variation analysis of malignant melanoma image yields a high count for number of significant color component bins

3.2.4 Texture Calculation

The texture variable, a measure of the lesion's color and textural variation, is the most novel algorithm developed in this software [3]. It utilizes the canny edge detection filter's edge detection to analyze the interior of the lesion. When the filter picks up edges, it creates white edges wherever there is a distinction between colors or other abrupt change in appearance. The percentage of white edges that are removed from within the lesion outer border then corresponds to the amount of variation in the lesion's appearance (and thus the crusty or flaky appearances of dangerous lesions). Therefore, texture was also a strong indication of malignancy.

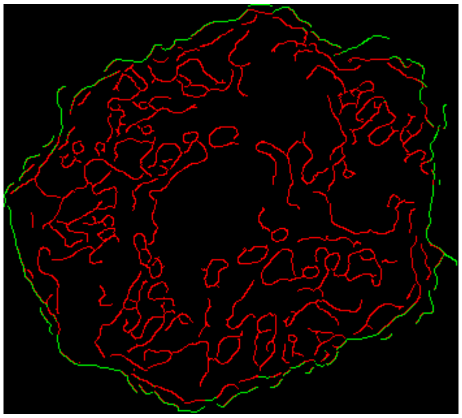


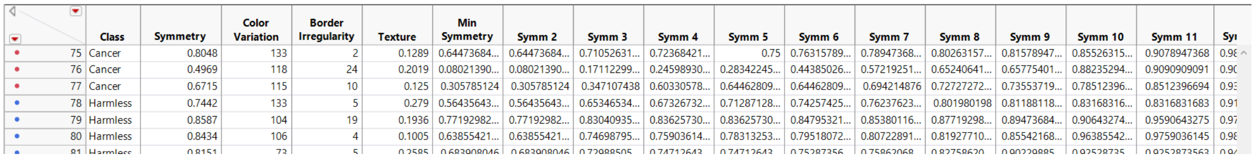
Figure 10. Texture analysis of malignant melanoma image. Green pixels represent segmented outer border, while red indicates large amount of detected texture, or variation

4. Multivariate Modeling, Variable Analysis, and Results

All the data for each variable were automatically extracted to a data table. Using discriminant analysis [2], the F-Ratio significance of all forty-four metrics were evaluated for all validation samples, and the best combination of variables (27 of 44) was selected to yield the lowest misclassification rate, 5.2%. The complete process was then automated in JSL.

4.1 Automated Metric Extraction

The values of all 44 variables for each sample are generated on one row of the data table. The first column in the table indicates the training data set's given class, and the rest of the columns represent extracted variables. This data table is then used for discriminant analysis.

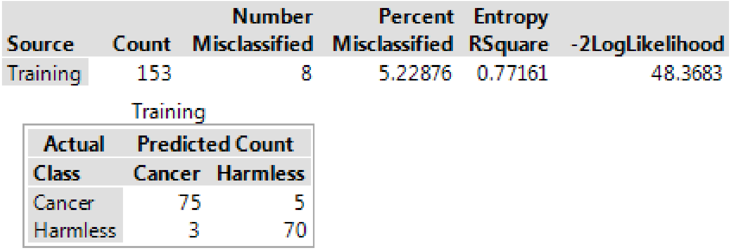


	Class	Symmetry	Color Variation	Border Irregularity	Texture	Min Symmetry	Symm 2	Symm 3	Symm 4	Symm 5	Symm 6	Symm 7	Symm 8	Symm 9	Symm 10	Symm 11	Sy
● 75	Cancer	0.8048	133	2	0.1289	0.64473684...	0.64473684...	0.71052631...	0.72368421...	0.75	0.76315789...	0.78947368...	0.80263157...	0.81578947...	0.85526315...	0.9078947368	0.98
● 76	Cancer	0.4969	118	24	0.2019	0.08021390...	0.08021390...	0.17112299...	0.24598930...	0.28342245...	0.44385026...	0.57219251...	0.65240641...	0.65775401...	0.88235294...	0.9090909091	0.90
● 77	Cancer	0.6715	115	10	0.125	0.305785124	0.305785124	0.347107438	0.60330578...	0.64462809...	0.64462809...	0.694214876	0.72727272...	0.73553719...	0.78512396...	0.8512396694	0.93
● 78	Harmless	0.7442	133	5	0.279	0.56435643...	0.56435643...	0.65346534...	0.67326732...	0.71287128...	0.74257425...	0.76237623...	0.801980198	0.81188118...	0.83168316...	0.8316831683	0.91
● 79	Harmless	0.8587	104	19	0.1936	0.77192982...	0.77192982...	0.83040935...	0.83625730...	0.83625730...	0.84795321...	0.85380116...	0.87719298...	0.89473684...	0.90643274...	0.9590643275	0.97
● 80	Harmless	0.8434	106	4	0.1005	0.63855421...	0.63855421...	0.74698795...	0.75903614...	0.78313253...	0.79518072...	0.80722891...	0.81927710...	0.85542168...	0.96385542...	0.9759036145	0.98
● 81	Harmless	0.8151	73	5	0.2585	0.683908046	0.683908046	0.72988505	0.74712643	0.74712643	0.75287356	0.75862068	0.82758620	0.90229885	0.92528735	0.9252873563	0.94

Figure 11. Data table generated from automated metric extraction program

4.2 Discriminant Analysis and Naive Bayes Modeling

The two classes used in this analysis are cancerous and harmless (in an alternative view, harmless lesions can be viewed as the control group), and the model predicts class membership and the probability of the lesion belonging in either class. The most accurate model used 27 variables (listed below) with the highest F Ratios (highest variation between classes and lowest variation within classes). Using this discriminant model, there was about a 5.2% misclassification rate, with a good model fit (Entropy RSquare of 0.77) [2].



		Number	Percent	Entropy	
Source	Count	Misclassified	Misclassified	RSquare	-2LogLikelihood
Training	153	8	5.22876	0.77161	48.3683

Training

Actual	Predicted Count	
Class	Cancer	Harmless
Cancer	75	5
Harmless	3	70

Figure 12. Model summary for discriminant analysis model that used 27 metrics, displaying the 5.2% misclassification rate, relatively high Entropy RSquare value (0.77), and distribution of incorrectly classified samples among cancerous and harmless lesions.

Variables used: symmetry, color variation, border irregularity, texture, minimum symmetry, 25% largest symmetry (3rd largest of 12), 58% largest symmetry (7th largest of 12), 75% largest symmetry (9th largest of 12), 92% largest symmetry (11th largest of 12), intensity variation, hue variation, redness variation, greenness variation, maximum intensity bin and corresponding index, maximum hue bin and corresponding index, maximum lightness bin and corresponding index, maximum saturation bin, maximum redness bin index, maximum greenness bin and corresponding index, maximum blueness bin index, minimum lightness bin index, minimum redness bin index, minimum greenness bin index

Throughout the subsequent figures, red represents cancerous lesions, while blue represents benign lesions. The discriminant analysis canonical plot is below, displaying the 152 samples and sample distribution with regard to the 50% confidence intervals (outer circles) and 95% confidence intervals (inner circles). The more different the two groups are, the less these ellipses overlap; therefore, the model identifies a clear distinction between the two groups.

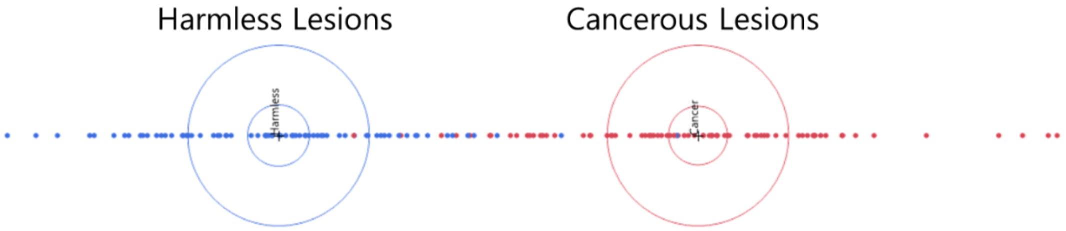


Figure 13. Discriminant analysis model classifies lesion images into malignant (red) and benign (blue) based on calculated metrics from algorithms, displaying sample distribution

The following parallel plot shows the 152 observations and their variable values relative to each other. Each line represents one observation and its corresponding values. There is a clear stratification between the cancerous and harmless lesion samples.

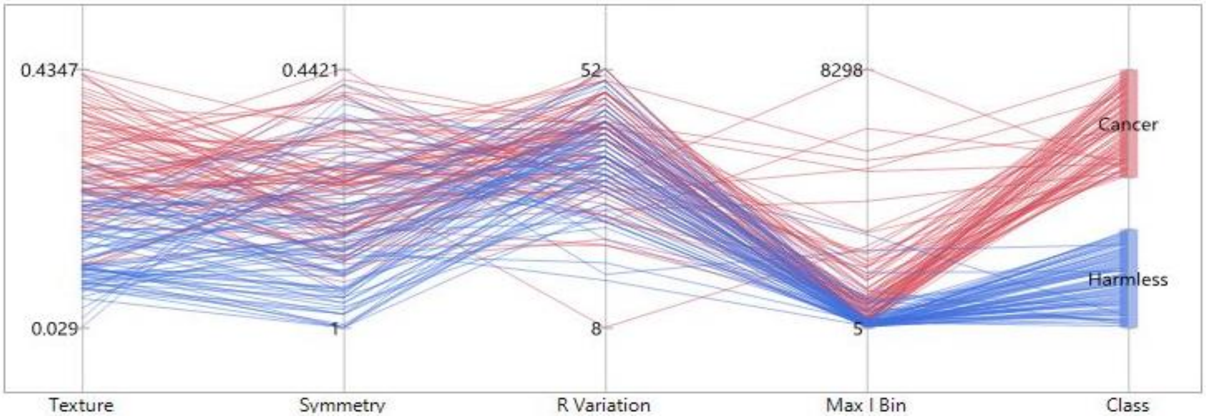
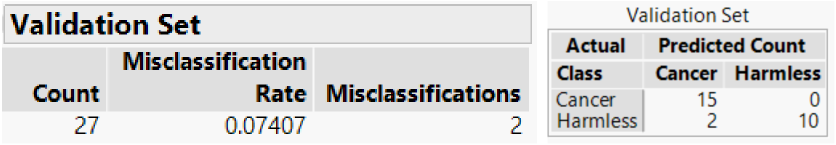


Figure 14. Parallel plot of 4 most significant variables in model that contribute to final lesion classification, showing clear distinction between two diagnosis classes

Cross-validation was then conducted with the 27 selected variables and a validation sample ratio of 0.10 (10%) using the Naive Bayes model [15]. This model achieved a 7.4% misclassification rate, validating the discriminant model. In addition, there were no false negative classifications, which is important for cautious diagnosis techniques.



Validation Set

Count	Misclassification Rate	Misclassifications
27	0.07407	2

Validation Set

Actual	Predicted Count	
Class	Cancer	Harmless
Cancer	15	0
Harmless	2	10

Figure 15. Cross-validation of discriminant analysis model using Naive Bayes model

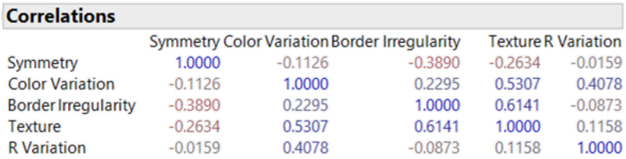
4.3 Variable Efficacy Analysis

The unpaired *t* test p values (N1 = N2 = 76) for the 10 most significant variables are listed below. These were low values ($p < 0.05$), indicating the variables were significant for separating the two classes of malignant and benign lesions.

<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0077	0.0259	0.0372
Texture	**75% symmetry**	**Intensity variation**	**Border irregularity**	**Max I Bin**	**Max R Bin Index**	**Min R Bin Index**	**Redness variation**

Figure 16. *t* test p values for 10 most significant variables in model and analysis

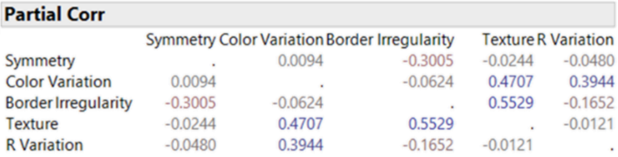
The correlations amongst the model's five most significant variables (the 4 primary variables, as well as one of the most significant sub-variables) were calculated. These variables showed little or no correlation with each other, demonstrating that each variable had a unique contribution to the classifications of the images.



Correlations

	Symmetry	Color Variation	Border Irregularity	Texture	R Variation
Symmetry	1.0000	-0.1126	-0.3890	-0.2634	-0.0159
Color Variation	-0.1126	1.0000	0.2295	0.5307	0.4078
Border Irregularity	-0.3890	0.2295	1.0000	0.6141	-0.0873
Texture	-0.2634	0.5307	0.6141	1.0000	0.1158
R Variation	-0.0159	0.4078	-0.0873	0.1158	1.0000

Figure 17. Correlation coefficients and partial correlation coefficients for the model's 5 primary variables (all values relatively low)



Partial Corr

	Symmetry	Color Variation	Border Irregularity	Texture	R Variation
Symmetry	.	0.0094	-0.3005	-0.0244	-0.0480
Color Variation	0.0094	.	-0.0624	0.4707	0.3944
Border Irregularity	-0.3005	-0.0624	.	0.5529	-0.1652
Texture	-0.0244	0.4707	0.5529	.	-0.0121
R Variation	-0.0480	0.3944	-0.1652	-0.0121	.

Figure 18. Partial correlation coefficients for 5 primary variables with all other variables (all values lower than correlation counterparts)

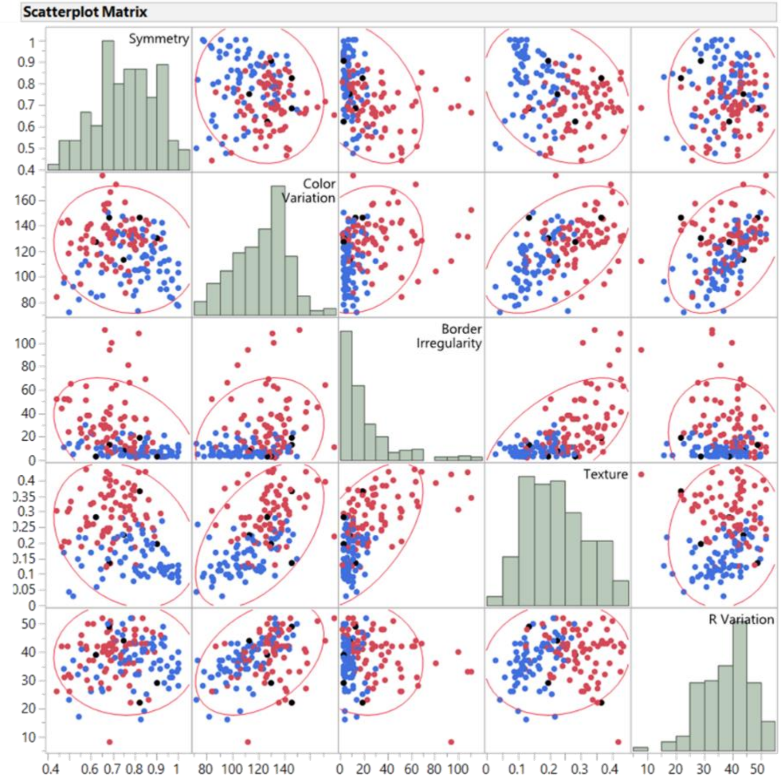
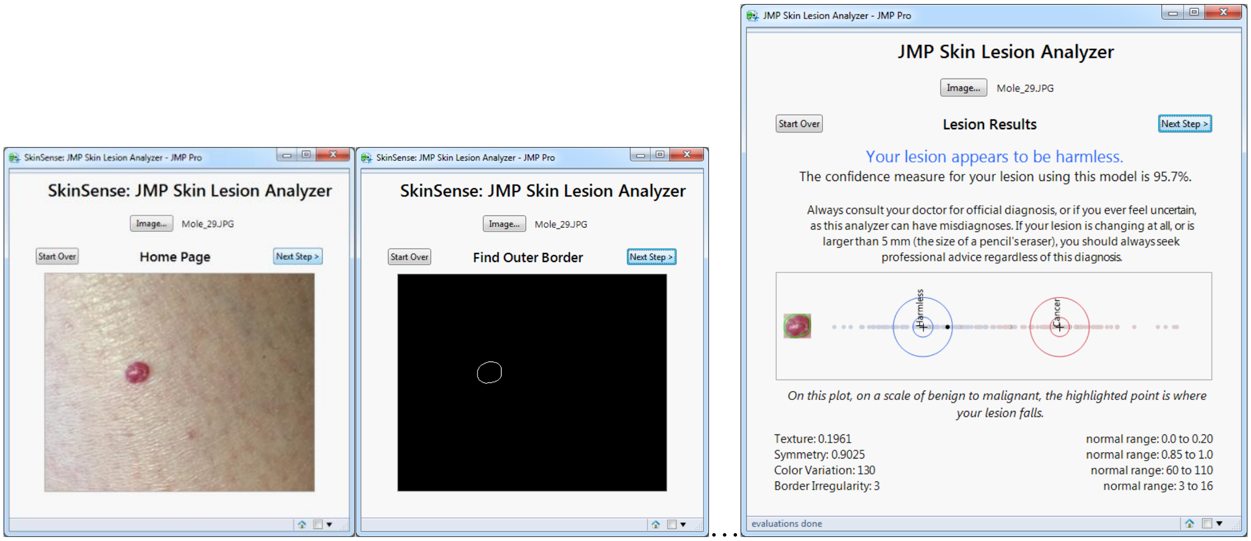


Figure 19. Correlation scatterplots and histograms for the discriminant model's 5 primary variables (symmetry, color variation, border irregularity, texture, and redness variation)

4.4 User Interface

A user interface was coded to allow for manual interaction with the automated software. This required manipulation of `H List Box` and `V List Box` commands to align display trees [10].

The interface allows the user to see all 10 steps of the image analysis and the lesion's final results.



Your lesion appears to be harmless.

The confidence measure for your lesion using this model is 95.7%.

Always consult your doctor for official diagnosis, or if you ever feel uncertain, as this analyzer can have misdiagnoses. If your lesion is changing at all, or is larger than 5 mm (the size of a pencil's eraser), you should always seek professional advice regardless of this diagnosis.

On this plot, on a scale of benign to malignant, the highlighted point is where your lesion falls.

Texture: 0.1961 normal range: 0.0 to 0.20
Symmetry: 0.9025 normal range: 0.85 to 1.0
Color Variation: 130 normal range: 60 to 110
Border Irregularity: 3 normal range: 3 to 16

Figure 20. User interface allows interaction with the automated software

The final page of the software shows where the user's lesion lies on the canonical plot, the prediction probability (confidence measure), and the primary metrics' values, as well as their normal ranges as determined by the training set. Creating the interface allows for the user to control the pace at which steps occur, even though the analysis requires no user input.

Overall, the software effectively and accurately offers improvements in the methods for automated analysis of skin lesions for cancer diagnosis.

5. Discussion

Overall, in this study, an interactive, automated software that accurately separated benign moles from cancerous growths was successfully created, and pertinent variables that contributed to a more accurate diagnosis were found. Lesion features were extracted and analyzed, and 27 metrics contributed to the final diagnosis model, which had a 5.2% misclassification rate, statistically significant metrics, and was successfully cross-validated with a Naive Bayes model.

Each step of the classification process has either made advances on original research, or has been confirmed as a reliable method. The segmentation algorithm in my software utilized the canny edge detector, a known effective, edge-based segmentation method. The texture variable I created through using the canny edge detection, however, has a more efficient, yet effective approach than previous texture calculations, which still aimed to detect surface variation, but used measures such as statistical moments [3]. The texture algorithm incorporates a second assessment of color variation, as the canny filter detects all changes in appearance in the lesion's interior, which includes variation in color. The developed noise removal algorithm also builds on previous efforts to remove hair or distracting features from lesion images [8, 14].

The symmetry algorithm used in this study also introduces rotational symmetry analysis, building on previous symmetry analyses that only took into account one orientation [5]. Taking individual symmetry measurements from different angles and sorting these values ascending to find patterns was novel. For example, asymmetrical cancerous lesions can appear symmetrical from certain angles (i.e. down the center of a lanceolate or oblong lesion). Calculating minimum symmetry was important to show symmetry extremes, which revealed a clear difference between the benign and cancerous classes. The same logic applies to the 25% largest symmetry metric, for example, which often delineated the point at which symmetry values diverged for the two classes, suddenly increasing for benign and continuing to be low for cancerous lesions.

The color variation algorithm developed is different from previous color segmentation algorithms in that each individual pixel is classified, and metrics extracted include indices of color classification, rather than producing a single variation score [17, 21]. This program allows individual components of color to be separately considered, and also searches for the major contributor of each component of color in the image, which generated multiple significant metrics, such as redness variation and maximum index.

Overall, given the low correlation coefficients of the variables used in this study, each had unique contributions to the statistical model, including the newly discovered variables that can be applied in future research. In addition, the discriminant analysis and cross-validation Naive Bayes models used in this study demonstrate that the lesion analysis variables developed in this study are effective, even more so than previous software.

Previous software have used expensive or relatively less accessible software for algorithm development, or required more difficult-to-acquire dermoscopy images [8, 14, 17]. This software did not have either of the two requirements, making it more accessible and implementable in other languages, such as R, Python, and Java. In addition, the automated analysis (as opposed to requiring user input) paves the way for future batch processing functions. As a more library-independent (thus more accessible) program, my software also communicates results interactively to the user, and can be used for screening by both patients and doctors.

As a whole, this skin detection software has been consistent with existing research and made advancements on multiple analysis algorithms, which was confirmed by statistical models.

6. Conclusion and Future Work

In this project, an interactive, automated software that accurately separated benign moles from cancerous growths was successfully created. Multiple novel, pertinent variables (not found in previous literature on the same topic with similar other methods [3]) and variable analysis algorithms that contributed to a more accurate diagnosis were found. The final statistical multivariate model had a 5.2% misclassification rate and statistically significant result, with a cross-validation misclassification rate of 7.4%, which can be effectively used as a preliminary screening by both patients and doctors.

Though this skin cancer detection software has already taken great strides in the medical image analysis field, there are areas for improvement. During sample collection, other information, such as size, evolution, itchiness, and family history could also be collected. Dermoscopy images could also be analyzed in addition to images collected via photography. Currently, lesions that are similar in color to the patient's skin are difficult to segment; auto-contrast adjustment could be added for low-contrast images, or a manual noise removal threshold slider could be added as an optional step in the interface. In border detection, K nearest neighbor methods could be implemented to fill in disconnected parts of the outer border. The symmetry algorithm could be improved to take into account the symmetry of the colors within the lesion, and not only the lesion's shape. The border irregularity calculation can be extended to extract more metrics, such as the maximum, minimum, and average magnitudes of slope change. A profit matrix can be added to the model to weight false positive diagnoses as more favorable than false negatives. Finally, a batch process functionality could be added.

There are many other applications for the methods used in this study, including analysis of other medical images (e.g. MRI, CT scans), handwritten character recognition, manufacturing defect detection (e.g. semiconductors), animal footprint recognition, facial recognition, fingerprint recognition, and topography analysis [7].

Note on personal role in project: I independently designed this project and sought guidance from my mentors (Principal Software Developer, Senior Product Manager) with regard to specific queries about syntax and statistical methods.

References

[1] "ABCD Rule Illustration." Wikipedia, Wikimedia Foundation, 20 July 2016, en.wikipedia.org/wiki/Melanoma.

[2] "Discriminant Analysis." *JMP*, SAS Institute, www.jmp.com/support/help/Disciminant_Analysis.shtml.

[3] Filho, Mercedes, et al. *A Review of the Quantification and Classification of Pigmented Skin Lesions: From Dedicated to Hand-Held Devices*. Porto.

[4] "Grayscale." *Wikipedia*, 31 Aug. 2016, en.wikipedia.org/wiki/Grayscale.

[5] Hossen Bhuiyan, Amran, et al. "Image Processing for Skin Cancer Features Extraction." *International Journal of Scientific & Engineering Research*, vol. 4, no. 2, Feb. 2013, kr.mathworks.com/matlabcentral/answers/uploaded_files/6298/Image%20Processing%20for%20Skin%20Cancer%20Features%20Extraction.pdf.

[6] "Images." *JMP 12 Online Documentation*, www.jmp.com/support/help/Images.shtml.

[7] "Image Segmentation." *Wikipedia*, 1 Sept. 2016, en.wikipedia.org/wiki/Image_segmentation#Applications.

[8] Kam Lee, Tim. *Measuring border irregularity and shape of cutaneous melanocytic lesions*. 2001. Simon Fraser University, PhD thesis. www.cs.sfu.ca/~stella/papers/2001/tim.thesis.pdf.

[9] Lounesto, Pertti. *Clifford Algebras and Spinors*. 2nd ed., Cambridge, Cambridge UP, 2001.

[10] "Manipulating Displays." *JMP*, www.jmp.com/support/help/Manipulating_Displays.shtml.

[11] "Melanoma, Brown and Red Lesion 1." *Wikipedia*, Wikimedia Foundation, 5 Mar. 2012, commons.wikimedia.org/wiki/File:Melanoma,_brown_and_red_lesion_1.jpg.

[12] "A Melanoma of Approximately 2.5 cm by 1.5 cm." *Wikipedia*, Wikimedia Foundation, 20 July 2016, en.wikipedia.org/wiki/Melanoma.

[13] "Melanoma, Red and Brown Lesion 2." *Wikipedia*, Wikimedia Foundation, 5 Mar. 2012, commons.wikimedia.org/wiki/File:Melanoma,_red_and_brown_lesion_2.jpg.

[14] Mishra, Nabin K., and M. Emre Celebi. *An Overview of Melanoma Detection in Dermoscopy Images Using Image Processing and Machine Learning*. Cornell University Library, 27 Jan. 2016, arxiv.org/ftp/arxiv/papers/1601/1601.07843.pdf.

[15] "Partition Method." *JMP*, www.jmp.com/support/help/K_Nearest_Neighbor.shtml.

[16] Ponte, John. "Image Analyzer." *User Community*, SAS Institute, 30 Jan. 2015, community.jmp.com/docs/DOC-7181.

[17] Rosado, Luís, and João M. Vasconcelos. "Pigmented Skin Lesion Computerized Analysis via Mobile Devices." *31st Spring Conference on Computer Graphics*, 22 Apr. 2015, pp. 105-08.

[18] Rosado, Luís, and Maria João M. Vasconcelos. *Automatic Segmentation Methodology for Dermatological Images Acquired via Mobile Devices*. SciTePress, 12 Jan. 2015.

[19] "Skin Cancer." *American Academy of Dermatology*, www.aad.org/media/stats/conditions/skin-cancer.

[20] "SkinVision." *SkinVision*, skinvision.com/.

[21] Umbaugh, Scott E., et al. *Automatic Color Segmentation of Images with Application to Detection of Variegated Coloring in Skin Tumors*.