AN AUTOMATED WOUND IDENTIFICATION SYSTEM

BASED ON

IMAGE SEGMENTATION AND ARTIFICIAL NEURAL NETWORKS

A Thesis
Submitted to the Faculty
of
Drexel University
by
Bo Song
in partial fulfillment of the requirements for the degree of
Master of Science
in
Biomedical Engineering
June 2012
ACKNOWLEDGEMENT

I would like to acknowledge all those who provided me with advice, support, and encouragement, with which can the completion of this thesis become possible.

I am sincerely indebted to my advisor, Dr. Ahmet Saçan, for his continuous support along the way. Without his guidance, trust, and encouragement, I would never have finished. I am also grateful to Dr. Andres Kriete and Dr. David E. Breen for serving on my thesis committee and for their valuable insights and comments.

I shall thank Tijo Abraham, Deepthi Vemanath, Kevin Alker, and Richard O. Ouma Jr., for their previous contributions from their senior design project; as well as all my colleagues in Sacan Bioinformatics Lab, for their supports and inspirations through our discussion about relevant knowledge, including Xiang Mao, Rehman Qureshi, Yiqian Zhou, Chunyu Zhao, Heng Yang, et al. My thanks also go to all my friends whose words and company meant a lot during the tough times, especially to Brian Erickson, who had also been a great help to my thesis writing and polishing.

Last but certainly not least, to my parents, Jianrong Song and Mei Chen, for their unconditional love, support, and understanding all the time, I dedicate my thesis.
# Table of contents

LIST OF TABLES .............................................................................................................. VI

LIST OF FIGURES ........................................................................................................... VII

ABSTRACT .......................................................................................................................... VIII

1. INTRODUCTION ............................................................................................................. 1

1.1. IMAGE SEGMENTATION ............................................................................................ 2

1.2. ARTIFICIAL NEURAL NETWORKS (ANNs) .................................................................. 3

   Development history ........................................................................................................ 4

2. METHODS ....................................................................................................................... 6

2.1. SELECTED SEGMENTATION METHODS ..................................................................... 7

   Thresholding ..................................................................................................................... 8

   K-means clustering ......................................................................................................... 8

   Edge detection ............................................................................................................... 9

   Region growing .............................................................................................................. 10

2.2. PRE-PROCESSING ..................................................................................................... 11

   Rescaling the image ....................................................................................................... 11

   Filtering the polygons .................................................................................................. 12

   Manual tracing ............................................................................................................. 13

   Performance criteria ................................................................................................... 14

2.3. PARAMETER OPTIMIZATION .................................................................................... 16

   Grid search ..................................................................................................................... 17

   Nelder-Mead simplex algorithm ................................................................................. 18

2.4. FEATURE EXTRACTION ............................................................................................ 19
2.5. ARTIFICIAL NEURAL NETWORKS (ANNs) .................................................................................. 22

Perceptron .................................................................................................................................................. 23

Learning Strategies ...................................................................................................................................... 27

Multi-Layer Perceptron (MLP) ..................................................................................................................... 28

Radial Basis Function (RBF) ......................................................................................................................... 29

3. EXPERIMENTS & RESULTS .................................................................................................................. 31

3.1. WOUND IMAGE COLLECTION ....................................................................................................... 31

3.2. TRAINING DATA PREPARATION ........................................................................................................ 32

3.3. DESIGNING THE NEURAL NETWORKS ............................................................................................... 36

3.4. TRAINING RESULTS .......................................................................................................................... 39

3.5. TESTING RESULTS ............................................................................................................................... 41

4. DISCUSSION & CONCLUSION .............................................................................................................. 45

LIST OF REFERENCES ............................................................................................................................... 49
List of Tables

Table 1. List of parameters ........................................................................................................... 18

Table 2. Algorithm of moment features of a polygon ................................................................. 20

Table 3. Composition of a feature vector ....................................................................................... 21

Table 4. Example of corresponding feature vectors and overlap scores (MCC) of the three picked segmented polygons shown in Figure 7 ........................................................................ 35

Table 5. Comparative performance of the two ANNs ................................................................. 43

Table 6. Comparative prediction with their ranking evaluation of the two ANNs ................. 44
List of Figures

Figure 1. Flow chart of the proposed method ................................................................. 6

Figure 2. Demonstration of the GUI for manual tracing. Multiple tracing tools can be selected to facilitate the process: a) poly; b) spline; c) closedline; d) rectangular; e) ellipse. 13

Figure 3. (a) Schematic representation of Matthews Correlation Coefficient (MCC); (b) Interpretation on wound image: area in color of yellow represents FN, green represents TP and orange represents FP. .............................................................................................................. 15

Figure 4. Illustration of biological neuron structure .......................................................... 23

Figure 5. Structure of an artificial neuron ........................................................................ 25

Figure 6. Structure of a typical three layer ANN ............................................................... 26

Figure 7. Hyperbolic tangent sigmoid activation function ............................................... 29

Figure 8. Gaussian function ............................................................................................ 30

Figure 9. GUI for manual trace of the wound region ....................................................... 33

Figure 10. Example of three picked segmented polygons by method of: a) K-means clustering; b) Edge detection; c) Thresholding ................................................................. 35

Figure 11. (a) MSE values over neuron numbers for different “spread” value. 40 as a number of neurons is where all the MSE values have a significant fall; (b) MSE values over “spread” values. Global minimum is achieved at “spread” of 230 ................................................. 38

Figure 12. (a) The structure of the designed MLP, (b) Training performance of the designed MLP ......................................................................................................................... 39

Figure 13. (a) The structure of the designed RBF, (b) Training performance of the designed RBF ......................................................................................................................... 40

Figure 14. The visualization results of segmented and filtered polygons by four segmentation methods on one test image. The desired polygon number is the second, tenth and the thirteenth (from left to right) ........................................................................................................ 42
**ABSTRACT**

An Automated Wound Identification System
Based on Image Segmentation and Artificial Neural Networks

Bo Song
Ahmet Saçan, Ph.D.

Chronic wounds are a global, ongoing health challenge that afflicts a large number of people. Effective diagnosis and treatment of the wounds relies largely on a precise identification and measurement of the wounded tissue; however, in current clinical process, wound evaluation is based on subjective visual inspection and manual measurements which are often inaccurate. An automatic computer-based system for fast and accurate segmentation and identification of wounds is desirable, both from the standpoint of improving health outcomes in chronic wound care and management, and in making clinical practice more efficient and cost-effective.

As presented in this thesis, we design such a system that uses color wound photographs taken from the patients, and is capable of automatic image segmentation and wound region identification. Several commonly used segmentation methods are utilized to obtain a collection of candidate wound areas. The parameters of each method are fine-tuned through an optimization procedure. Two different types of Artificial Neural Networks (ANNs), the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF) with parameters decided by a cross-validation approach, are then applied with supervised learning in the prediction procedure, and their results are compared. Satisfactory results of this system suggest a promising tool to assist in the field of clinical wound evaluation.
CHAPTER 1

1. INTRODUCTION

The treatment and management of chronic wounds is an ongoing health challenge affecting approximately 5.7 million patients in the US [1], especially the elderly and bedridden. Identifying the wound area is the first step in assessment and treatment of chronic wounds. However, in clinical practice, wound evaluation is mostly based on visual inspections and hands-on measurements which are subjective and inaccurate [2]. Digital image based and computer aided segmentation have proved effective in improving the accuracy of wound assessment [3].

However, the parameters of an image segmentation method differ from image to image, and are always decided by prior knowledge or from a large amount of trial experiments. Moreover, existing generic methods for image segmentation are themselves task-agnostic especially when applied to domain-specific problems [4]. When it comes to the wound identification application, semiautomatic methodology has performed well [5], but it remains labor intensive.

This paper proposes automated methods for wound segmentation and the subsequent identification. Parameter optimization is employed to automatically fine-tune the segmentation for specific samples in the domain; neural networks are then employed to learn to identify and evaluate the segmentation results automatically.
1.1. Image Segmentation

Image segmentation is usually a necessary and the first step of identification of objects. With high quality results provided by the segmentation, can the identification procedure works effective and efficient [6].

The early attempt and effort of studying the algorithms of image segmentation has been focusing on gray scale (monochrome), and more recent research has extended this development to color image applications [7]. Due to the solid underlying foundation based on the achievements in the areas of mathematics, statistics, and physics, large amount of image segmentation algorithms emerge with considerable success in the field. This is particularly true after the inventions of the advanced computer and memory technology which give strong supports to faster data processing and make more powerful algorithms for image segmentation possible.

General classification of these numerous image segmentation algorithms falls into three categories: Pixel-based algorithms, edge-based algorithms, and region-based algorithms. Pixel-based algorithms, as its name implies, focus mainly on the intensity or color of individual pixels, edge-based algorithms interpret the discontinuities between regions in the image, while the region-based algorithms focus on the opposite, which is the regions of continuity [8]. Several image segmentation methods are discussed in more detail in Section 2.1.
As these three categories are classified by the different inherent features of an image that are utilized for the algorithm, the efficacy of them specifically depends on the certain characteristics of the images they are applied to.

1.2. Artificial Neural Networks (ANNs)

In order to utilize the outcomes from the image segmentation, Artificial Neural Networks (ANNs) are the popular technique to be considered for the task of identification that needed to be accomplished automatically by machine.

ANNs are the approaches in the study of Machine Learning, and Machine Learning is an essential branch of Artificial intelligence (AI) [9]. Biologically inspired by the information processing mechanism and functionality of the human brain, ANNs gain partly the intelligent features and a rapidly growing interest in the field, as its simplified but massively parallel distributed topology possesses great advantages of massive parallelism, robustness and approximate reasoning. ANNs are often effective at problems that are difficult to process through sequential computational with conventional approaches, but which are easily solved by human beings, such as identification, pattern recognition, classification, data prediction, decision making and generalization. Success of the ANNs has been proved particularly in "fuzzy" applications where information may be incomplete or ambiguous, including applications in medicine (diagnosis and analysis), engineering, physics and others [6, 9].
In order to utilize ANNs for solving a problem solving, several general steps can be followed [10, 11]:

1. Problem identification. Identify the generic problem and the kind of information is available.
2. Choosing the appropriate ANN to solve the problem.
3. Preparing data for training the network.
4. Training a neural network when data for training are available.
5. Testing the generalization ability of the trained ANN and validating the results.

**Development history**

The development of ANNs has waxed and waned through its long history [12, 13] which can be dated to hundreds of years ago, but the arguably formative start of the modern era began with a classic paper wrote in 1943 by neuroanatomist Warren McCulloch and mathematical prodigy Walter Pitts where the first formal computing model of an artificial logical neuron and network is created and significantly establish the discipline of Artificial intelligence (AI). This was followed in 1949 by a book entitled "The Organization of Behavior" by Donald Hebb, describing a fundamental rule, known as Hebbian Learning Law, for neural system learning and training of a network. In 1958, fifteen years after the 1943 paper, a term "perception" was introduced by Frank Rosenblatt at Cornell University, and with this model certain classifications with supervised learning when the input space is linearly separable is able to be successfully handled.
In spite of the early success of the primitive perception, its limitation was considered as insurmountable from 1969, and ushered in decades of dormancy where the research of ANNs attracted a minimum of interest. This period ended in the 1980s when several major developments in ANN theory and design, such as the rediscovery of backpropagation training algorithm, led back the widespread interest and a continuously explosive development in the field of ANNs.
Chapter 2

2. METHODS

A summary of the method proposed in this thesis for automated wound identification is illustrated in Figure 1.

Figure 1. Flow chart of the proposed method
During the training procedure, wound images in the database are segmented by different methods and their corresponding optimized parameters; the resulting polygons are then preprocessed by a filter to remove those that are too large or small, and reduce the quantity of polygons for faster processing in the following steps. In every image, shape and region properties are extracted from each of the segmented and filtered polygons to form their feature vectors; simultaneously, each polygon is compared with the manually traced wound area to obtain an overlap score. Feature vectors and overlap scores are then passed through the neural networks to train the identification system.

During the operation procedure, the wound area in any new wound image will be automatically recognized and evaluated by the trained system based on feature vectors of segmented and filtered polygons of that image.

In the following sections, each particular step of the above algorithm will be described in detail.

2.1. Selected Segmentation Methods

Accurate segmentation of the wound image is one of the fundamental precepts in numerous fields such as image analysis and pattern recognition, and a primary prerequisite in our system for successful automatic identification of the wound area. However, due to the variety type of wounds with different features presented in a digital image, a precise automatic segmentation is not trivial. Although image
segmentation has been studied for decades [8, 14], there is still not a universal method in existence that can be applied to all types of images with diverse features [15].

For our system, we take four popular image segmentation methods into consideration: Thresholding, k-means clustering, edge detection, and region growing.

**Thresholding**

The thresholding method relies on an intensity value, which is called the “threshold”, to determine different classes that a pixel should belong to. It is one of the oldest and most popular pixel-based algorithms.

In the simplest case, two distinct objects in the image will present different peaks in its histogram (the histogram is calculated from all the pixels intensity in the image). Each peak represents one of the objects and is separated by a valley. Based on this feature of the image, a threshold setting at the bottom of the valley will always distinguishes the two objects, by grouping pixels with intensities greater and lesser than the threshold into different classes [8, 16].

**K-means clustering**

Clustering is another very popular pixel-based algorithm, and among which K-means clustering algorithm received extensive attention. The K-means clustering algorithm partitions an image into K clusters through iterations.

This algorithm starts by setting K predefined initial cluster centroids. Within each iteration, every pixel in the image will then be compared with all the centroids, using the Euclidean distance as the criterion. Every pixel will be assigned to the
cluster with a minimum pixel-to-centroid distance. After the assignment of all the pixels, new cluster centroids will be calculated by the mean intensity of all the pixels in each cluster. With the new centroids, the iteration continues until the new centroids are same as the centroids in the last iteration, representing the convergence of the algorithm [8, 17].

**Edge detection**

Edge detection utilizes the features on the boundaries of different regions which are defined as sharp changes in intensities. These sharp changes are characterized through and can be detected by using first order derivative or second order derivative.

As in this gray scale example image, the result of the first order derivative presents zero in the place of no changes in intensities, but positive at the ascending edge of the transition, while negative at the descending edge. The second order derivative, as the derivative of the first order derivative, will present a zero crossing at the center of the edge in the image.

The expression applied for the first order derivative is usually the gradient magnitude while that for the second order derivative is generally the Laplacian. Based on these two principles of edges detection, different type of edge operators with their own advantages can be applied to approximate the derivatives. Well-developed edge operators in existence are: Sobel, Prewitt, and Roberts for the first order derivative, and Laplacian of a Gaussian (LoG), Zero-crossings, and Canny for the second order derivative [8, 18].
**Region growing**

In the region growing approach, an initial pixel is selected as a “seed” point, and as its name implies, a region will grow around this seed until a certain criterion is reached, and hence the region is finally generated.

The criterion is used iteratively to examine the neighboring pixels of the seed, or later, the boundaries of the region. If the criterion is satisfied, the neighboring pixels will be included in the existing region and the region will be updated; otherwise they are excluded. When no new neighboring pixels can be included in the region, the iteration will end and the region is eventually segmented out [8, 19].

Each segmentation method is designed and developed for a particular type of image whose features and conditions are specific, hence none of these popular methods for image segmentation is considered effective for all images; in addition, not all methods perform equally well for the same image [15]. In order to tackle this problem, we pack these four selective segmentation methods as one integrated segmentation approach, and parameters of each method will be regulated for one specific type of wound image, which greatly increasing the robustness of the system. Under this solution, each of the distinct types of wounds can always be effectively segmented by at least one segmentation method from the integrated segmentation approach.
Furthermore, while all of the methods have been well-developed in gray-scale, we have applied them in color space to allow more information to be utilized and hence generate more preferable results [7].

2.2. Pre-Processing

In order to guarantee the segmentation procedure and the later identification process be more effective, the image or data in the preceding and succeeding stages of the segmentation will need to be preprocessed and prepared.

Rescaling the image

In case of the images that need to be deal with are of large size with massive pixels, rescaling should be considered as a way to improve the processing speed for the image segmentation. This is particularly the case for the pixel-based segmentation methods. Our system uses wound images taken from clinical environment by commercial cameras, which nowadays are commonly featured with highly advanced MEGA-pixel digital Charge-Coupled Device (CCD) or Complementary Metal–Oxide–Semiconductor (CMOS) sensor. Thus we apply adaptive rescaling on the images prior to selected segmentation methods. The parameter of our rescaling method is “scale”, which is listed in the Table 1 (the list of parameters).

The value of “scale” may be a positive real number between 0 and 1, in which case the “scale” is a contraction ratio to shrink the original image; or it may be a negative integer number that represents the desired pixels of the maximum image dimension (width or height, automatically chosen) of the contracted image. In both cases, the aspect ratio of the dimensions of the original image is maintained.
The rescaled images for the later segmentation contain fewer pixels and hence will greatly reduce the time required for the segmentation procedure and hence improve the system. After the segmentations of the rescaled images, the resulting polygons will be mapped back to the original large image.

**Filtering the polygons**

Due to the fact that each of the four segmentation methods may generate numerous polygons for one image, the integrated segmentation run for all the images in the database will get a considerable number of polygons. As the quantity of segmented polygons has a direct impact on the performance of the system, eliminating trivial polygons will reduce the following processing amount. We create a filter to reduce the number of segmented polygons.

The criteria of the filter are based on occupancies and locations of the polygons in the image. Polygons that possess too large or too small areas in relation to the total size of the image, as well as polygons that located too close to the border of the image, will be considered trivial (highly unlikely to be the right wound region) and be eliminated by the filter. The filtering procedure makes the later procedure more meaningful and effective.

The parameter “minarea” for the filter has a real value as a criterion to determine the polygons with polygon-to-image ratio less than the “minarea” to be too small, while polygons are determined as too large if their polygon-to-image ratios are larger than the value of filter’s parameter “maxarea”. These two parameters of the filter are also shown in the Table 1 as a part of the list of parameters.
Manual tracing

In order to provide performance criteria for the training of the ANNs, as well as the optimization of parameters for the segmentation methods, a manually traced wound region is required.

We create a graphical user interface (GUI) for the purpose of manual. The GUI presents a wound image on a computer screen; clinical experts can then use a mouse to trace the boundary of the wound based on their clinical experiences. The visualized image can be zoomed for detail and precise boundary tracing. Multiple tracing tools are available to facilitate the tracing process and their effects are demonstrated in Figure 2.

![Figure 2](image)

Figure 2. Demonstration of the GUI for manual tracing. Multiple tracing tools can be selected to facilitate the process: a) poly; b) spline; c) closedline; d) rectangular; e) ellipse.
Performance criteria

For every polygon resulting from the segmentation of an image, a performance criterion is calculated for evaluation, optimization and ANNs training. Generated polygons from the segmentation for each image are compared with the manually traced wound region of that image by the Matthews Correlation Coefficient (MCC) measure algorithm to score the agreement between the manually traced and automatic segmented versions. The resulting overlap score of each polygon is then used as its performance criterion.

The Matthews Correlation Coefficient is often used for two-class classifications and is considered an effective performance metric [20]. It takes true and false positive and negative values from observed and predicted classifications to calculate a score between -1 and +1, where -1 represents a worst prediction, 0 means no better than random and +1 indicates the best agreement between two classification. MCC can be represented by the formula:

\[
\text{MCC} = \frac{(TP \times TN)-(FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}
\]  

(1)

Where, TP (true positive) represents the pixels of overlapping between P (manually traced wound region) and Q (segmented polygon), TN (true negative) represents the pixels that do not belong to either P or Q, FP (false positive) represents the pixels belonging to Q but not to P, and FN (false negative) represents the pixels in P but not Q. Graphic interpretations of the MCC calculation is shown in Figure 3.
As we can conclude, the higher the overlap score of a polygon, the better the chance of it being considered a representative of the expected wound region.

An alternative measure algorithm named “F-measure” was also considered, which performs similarly to the MCC for the same situation. The results of F-measure are not shown here, as we decide to use MCC to be the performance criteria for the segmented polygons.

Figure 3. (a) Schematic representation of Matthews Correlation Coefficient (MCC); (b) Interpretation on wound image: area in color of yellow represents FN, green represents TP and orange represents FP.
2.3. Parameter Optimization

The parameters of a segmentation method are a critical factor in the results obtained, and different segmentation methods are optimal for different images with different features. Thus, the parameters of each segmentation method need to be carefully determined with respect to the different types of wound images they are designed to handle.

The parameter $K$, which defines the number of clusters in K-means clustering segmentation, is a good example of this necessity. The value of $K$ has a significant effect on the segmentation results. If a proper value of $K$ is large enough to generate effective segmentation for a given wound of small size, it will over-segment or not satisfactorily segment a wound with relatively large size, and vice versa. Conventionally, selecting a $K$ requires prior knowledge or trial experiments; it was not very accessible especially when considering the interactions of $K$ with other parameters of the method, and it is difficult to implement quantifiable evaluations for the consequence as well. This thorny issue exists for all other parameters in every segmentation methods.

As a solution, we propose a procedure to automatically optimize the parameters of each segmentation method. In the optimization procedure, the images with similar features will be grouped and segmented by one of the four segmentation methods with parameters in their initial values. This will generate multiple polygons, which are then compared to manually traced wound regions of the corresponding images to produce overlap scores. The mean value of all the highest scores collected
from each image in one group is used as an optimization criterion for the certain segmentation method. This process will be repeated iteratively with altered parameters until the minimum of optimization criterion is found. The corresponding parameters will then be used as the optimal parameters for the application of the particular segmentation method and image type. This automatic parameter optimization process generally guarantees accurate segmentation for different type of wound images.

As there are total of two data types, integer and continuous non-integer, for the segmentation parameters (detailed in Table 1), the parameter search process is handled by a combination of two methods, Grid search and the Nelder-Mead simplex algorithm.

**Grid search**

Grid search is a well-known method for searching the optimal values. It constructs grids within the allowed range of the bounded variables, and evaluates the function at each grid point. The global minimum is returned as the optimal result. The Grid search method is a simple and often effective means in searching for optimal parameters, but can carry a high computational cost related to the number of grid points [21] as its search method is exhaustive. Thus, we apply Grid search only to integer-based segmentation parameters, and employ another optimization method named the Nelder-Mead simplex algorithm for continuous non-integer parameters.
**Nelder-Mead simplex algorithm**

The Nelder-Mead simplex algorithm is a nonlinear optimization technique which is both effective and computationally compact. It uses the term simplex, and the function will be evaluated at its vertices, which are iteratively replaced by new best points. Thus, the simplex is shrunk towards the optimum until the best value of the function is found [22].

With the parameter optimization method we proposed, the parameters of all segmentation methods for different types of wound images will be optimized automatically. The optimized parameters will then be saved as default values for the automated system.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Parameters</th>
<th>Data type</th>
<th>Default value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thresholding</td>
<td>thresholddist</td>
<td>real</td>
<td>0.1</td>
</tr>
<tr>
<td>K-means clustering</td>
<td>nclusters (K)</td>
<td>integer</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>posweight</td>
<td>real</td>
<td>0.03</td>
</tr>
<tr>
<td>Edge detection</td>
<td>fudge</td>
<td>real</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td>erosionsize</td>
<td>integer</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>edgemethod</td>
<td>integer</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>strellinelen</td>
<td>integer</td>
<td>2</td>
</tr>
<tr>
<td>Region growing</td>
<td>seedcolor</td>
<td>integer</td>
<td>[142.1339, 45.9967, 35.4097]</td>
</tr>
<tr>
<td></td>
<td>maxdist</td>
<td>real</td>
<td>0.15</td>
</tr>
<tr>
<td>Rescaling</td>
<td>Contraction</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>scale</td>
<td>integer</td>
<td>-300</td>
</tr>
<tr>
<td>Filter</td>
<td>Occupancy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>minarea</td>
<td>real</td>
<td>1/2500</td>
</tr>
<tr>
<td></td>
<td>maxarea</td>
<td>real</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>Location</td>
<td>mindistfromborder</td>
<td>real</td>
</tr>
</tbody>
</table>
2.4. Feature Extraction

Following image segmentation with optimized parameters, all the resulting polygons will be filtered and then used to train the neural networks. In order to prepare each polygon for the neural networks, we represent them as feature vectors.

We extracted forty-nine features out of each segmented and filtered polygon to form a representative feature vector. Each feature vector consists of the information of geometry measurements (boundary integrals for area, centroid and area moment of inertia), shape measurements and the pixel value measurements of the polygon that it represents.

The extraction of moments of inertia from a polygon uses the algorithm proposed by H.J. Sommer III [23], and the polygonal approximation expression of which is shown in the Table 2. Shape and the pixel value measurements of a polygon are extracted by the association of their masks (logical matrix) and intensities.

The details of all forty-nine features that compose the feature vector of a polygon are listed in the Table 3 with the length and description of each feature.

All the feature vectors extracted from the segmented and filtered polygons will be used as training inputs. Meanwhile, their corresponding overlap scores will be calculated by MCC measure and be used as desired outputs for the training of the prediction system.
Table 2. Algorithm of moment features of a polygon

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Clockwise Summations for Closed Polygon</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( n ) vertices ( x_i, y_i )</td>
</tr>
<tr>
<td></td>
<td>( \Delta x = x_{i+1} - x_i )</td>
</tr>
<tr>
<td></td>
<td>( \Delta y = y_{i+1} - y_i )</td>
</tr>
<tr>
<td></td>
<td>( x_{n+1} = x_1 ) \quad ( y_{n+1} = y_1 )</td>
</tr>
<tr>
<td>First Moment</td>
<td>( A_{xc} ) about ( y ) axis</td>
</tr>
<tr>
<td>( n ) \sum_{i=1}^{n} \frac{1}{12} (6x_i y_i \Delta x - 3x_i^2 \Delta y + 3y_i \Delta x^2 + \Delta x^2 \Delta y ) }</td>
<td></td>
</tr>
<tr>
<td>First Moment</td>
<td>( A_{yc} ) about ( x ) axis</td>
</tr>
<tr>
<td>( n ) \sum_{i=1}^{n} \frac{1}{12} (3y_i^2 \Delta x - 6x_i y_i \Delta y - 3x_i \Delta y^2 - \Delta x \Delta y^2 ) / 12</td>
<td></td>
</tr>
<tr>
<td>Second Moment</td>
<td>( I_{xx} ) about ( x ) axis</td>
</tr>
<tr>
<td>( n ) \sum_{i=1}^{n} \frac{1}{12} (2y_i^3 \Delta x - 6x_i y_i^2 \Delta y - 6x_i y_i \Delta y^2 - 2x_i \Delta y^3 - 2y_i \Delta x \Delta y^2 - \Delta x \Delta y^3 ) / 12</td>
<td></td>
</tr>
<tr>
<td>Second Moment</td>
<td>( I_{yy} ) about ( y ) axis</td>
</tr>
<tr>
<td>( n ) \sum_{i=1}^{n} \frac{1}{12} (6x_i y_i^2 \Delta x - 2x_i^3 \Delta y + 6x_i y_i \Delta x^2 + 2y_i \Delta x^3 + 2x_i \Delta x^2 \Delta y + \Delta x^3 \Delta y ) / 12</td>
<td></td>
</tr>
<tr>
<td>Cross Moment</td>
<td>( I_{xy} )</td>
</tr>
<tr>
<td>( n ) \sum_{i=1}^{n} \frac{1}{24} (6x_i y_i^2 \Delta x - 6x_i^2 y_i \Delta y + 3y_i^2 \Delta x^2 - 3x_i^2 \Delta y^2 + 2y_i \Delta x^2 \Delta y - 2x_i \Delta x \Delta y^2 ) / 24</td>
<td></td>
</tr>
<tr>
<td>Centroidal moments</td>
<td>( I_{uu} )</td>
</tr>
<tr>
<td>( I_{uu} = I_{xx} - A y_c^2 )</td>
<td></td>
</tr>
<tr>
<td>Centroidal moments</td>
<td>( I_{uv} )</td>
</tr>
<tr>
<td>( I_{uv} = I_{yy} - A x_c^2 )</td>
<td></td>
</tr>
<tr>
<td>Centroidal moments</td>
<td>( I_{uv} )</td>
</tr>
<tr>
<td>( I_{uv} = I_{xy} - A x_c y_c )</td>
<td></td>
</tr>
<tr>
<td>Centroidal polar moment</td>
<td>( J )</td>
</tr>
<tr>
<td>( I_{uu} + I_{vv} )</td>
<td></td>
</tr>
<tr>
<td>Centroidal principal moments</td>
<td>( I_1 )</td>
</tr>
<tr>
<td>( I_1 = \frac{(I_{uu}+I_{vv})}{2} + \sqrt{\frac{(I_{uu}-I_{vv})^2}{4} + I_{uv}^2} )</td>
<td></td>
</tr>
<tr>
<td>Centroidal principal moments</td>
<td>( I_2 )</td>
</tr>
<tr>
<td>( I_2 = \frac{(I_{uu}+I_{vv})}{2} - \sqrt{\frac{(I_{uu}-I_{vv})^2}{4} + I_{uv}^2} )</td>
<td></td>
</tr>
</tbody>
</table>
Table 3. Composition of a feature vector

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Length</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC</td>
<td>2</td>
<td>image dimension</td>
</tr>
<tr>
<td>colormax</td>
<td>1</td>
<td>color scale of the image</td>
</tr>
<tr>
<td>area</td>
<td>1</td>
<td>Boundary integrals for area</td>
</tr>
<tr>
<td>centroid</td>
<td>2</td>
<td>Boundary integrals for centroid</td>
</tr>
<tr>
<td>perimeter</td>
<td>1</td>
<td>Boundary integrals for perimeter</td>
</tr>
<tr>
<td>$I(x&amp;x)$</td>
<td>3</td>
<td>second moment of polygon Inertia</td>
</tr>
<tr>
<td>$I(u&amp;v)$</td>
<td>3</td>
<td>centroidal moments</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>centroidal polar moment</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
<td>centroidal principal moments</td>
</tr>
<tr>
<td>ang</td>
<td>2</td>
<td>angles moment in radians</td>
</tr>
<tr>
<td>Area</td>
<td>1</td>
<td>number of pixels in the polygon</td>
</tr>
<tr>
<td>Centroid</td>
<td>2</td>
<td>center of mass of the polygon</td>
</tr>
<tr>
<td>BoundingBox</td>
<td>4</td>
<td>smallest rectangle containing the polygon</td>
</tr>
<tr>
<td>MajorAxisLength</td>
<td>1</td>
<td>length of the major axis of the polygon</td>
</tr>
<tr>
<td>MinorAxisLength</td>
<td>1</td>
<td>length of the minor axis of the polygon</td>
</tr>
<tr>
<td>EulerNumber</td>
<td>1</td>
<td>Euler number of the polygon</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>1</td>
<td>eccentricity of the polygon</td>
</tr>
<tr>
<td>Orientation</td>
<td>1</td>
<td>angle between the x-axis and the major axis of the polygon</td>
</tr>
<tr>
<td>FilledArea</td>
<td>1</td>
<td>area of hole –filled polygon</td>
</tr>
<tr>
<td>Solidity</td>
<td>1</td>
<td>proportion of pixels in convex hull as well as in polygon</td>
</tr>
<tr>
<td>Extent</td>
<td>1</td>
<td>ratio of pixels in polygon to pixels in total bounding box</td>
</tr>
<tr>
<td>Perimeter</td>
<td>1</td>
<td>distance around the boundary of the polygon</td>
</tr>
<tr>
<td>WeightedCentroid</td>
<td>6</td>
<td>centers of the polygon based on locations and intensities</td>
</tr>
<tr>
<td>MeanIntensity</td>
<td>3</td>
<td>mean of all the intensities in 3 color channel values in polygon</td>
</tr>
<tr>
<td>MinIntensity</td>
<td>3</td>
<td>the greatest intensities in 3 color channel in polygon</td>
</tr>
<tr>
<td>MaxIntensity</td>
<td>3</td>
<td>the lowest intensities in 3 color channel in polygon</td>
</tr>
</tbody>
</table>
2.5. Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANNs) gained great popularity for their adaptability and success in solving real world problems, especially those with no known description functions [24], and draw their strength from simulation of the processing structure of the human brain. The tremendous information processing capability of the brain relies on its massively parallel distributed structure and functional transmission of information. Its fundamental functional unit is the nerve cell, also known as neurons. ANNs consist of nodes representing neurons, which are interconnected to form a network with a powerful capability for parallel computing and nonlinear generalization similar to the human learning process [11].

It is estimated that human brain contains approximately 100 billion ($10^{11}$) neurons and nearly 100 trillion ($10^{14}$) interconnections participating in the transmissions of signals and information. Although individual neurons are very simple processors, which are 5 or 6 orders of magnitude slower than silicon logic gates in the chip of the computer, the brain as a whole can process information enormously more efficiently and intelligently than the most powerful computer.

In order to take use of the advantages of human brain, “perceptron” is created to mimic the performance of the neuron and used as a basic functional structure to form layers of topology in different types of neural network which can take supervised learning and hence gain the “intelligence”.
Perceptron

A schematic diagram of the anatomical structure of biological neurons and the intercommunication among them is shown in Figure 4.

Figure 4. Illustration of biological neuron structure
As the diagram [25] indicates, a neuron is composed of synapses, dendrites, an axon, and a cell body. The transmission of electrical signals within a neuron occurs as follow [26]:

1. Signals are received from other neurons through the thousands of dendrite branches;
2. The cell body receives these incoming signals which are then integrated in a certain way;
3. If the integrated signal exceeds a certain threshold value, it triggers the generation of an impulse from the cell body which is conducted down the axon that connects to the dendrites of other neurons (or in some cases, directly to the cell bodies). When a neuron generates this impulse or “action potential”, it is said to be activated.

Junctions between two individual neurons are called synapses. The connections of synapses play an important role as they are able to transfer the incoming signals to be excitatory factors or inhibitory factors towards the activation of the cell body.

The “perceptron” is a concept that is built base on the single artificial logical node to model the biological neuron. For this reason, perceptron is now also named as artificial neuron. The structure of a perceptron, or an artificial neuron, can be depicted as Figure 5.
The structure of the artificial neuron bears resemblance to a biological neuron, and works similarly [11]: synapses are modeled as “weights” to affect the strength of excitatory or inhibitory effect. The weighted inputs travel to the neuron node and are integrated by the input function. An activation function compares the result of integration to a threshold, and activates if the value is great enough. Its activated signal is then sent with outgoing weights to each of its connected neuron.

A neural network is composed of many artificial neurons in different topologies; neurons with similar purposes are grouped in layers. A typical formation of a three-layer ANN with feed forward architecture is illustrated in Figure 6.
In a typical three-layer ANN, the first layer is also called “input layer” and receives incoming signals. Each input value is then transmitted, through weights, to the neurons that consist of the middle layer, which is also called the “hidden layer”. Each of the neurons in the hidden layer has an associated activation or mapping function that will transform weighted inputs and deliver the final outputs to the “output layer” [11].
Learning Strategies

One of the essential properties of the human brain, which ANNs particularly attempt to emulate, is the capability to build rules through experience: to learn from surrounding environment. This is the property that allows ANNs to solve ambiguous problems that traditionally require human intelligence. ANNs must be stimulated by the environment to build rule sets, and corresponding changes taken place in the internal structure will produce new and more accurate responses to the environment. When the network responds accurately to stimulation, it is referred to as “trained.” The two most common learning strategies being used to train ANNs are supervised learning and unsupervised learning.

Supervised learning, or learning with an imaginary teacher, is an approach where the input vector drawn from the input space (environment) is associated with a desired output (response) provided by the teacher to train the network. For example, given an input-output (desired) associated pair \((x, y)\), where \(x\) represents the input vector from input space, and \(y\) is the desired output provided, the network will produce its own output denoted as \(f(x)\). When \(f(x)\) is compared against \(y\), their difference can be described by the expression:

\[
error = y - f(x) \tag{2}
\]

where “error” represents the approximation error according to which the network will be changed. During the learning procedure, the network is configured until the optimum change which minimizes the error function is reached [11].
Unsupervised learning, on the other hand, has no such desired output $y$ provided. During the learning process, the network modifies itself according to the given inputs without a teacher. It attempts to discover statistically salient features, characteristics, concepts, clusters etc. from the inputs, and classify their similarities [11].

Pursuant to wound identification as proposed in this research project, we choose two types of ANNs, both of which are popular and use the perceptron as the basis of their feed forward, supervised learning networks. The first one is Multi-Layer Perceptron (MLP) and the other is the Radial Basis Function (RBF).

**Multi-Layer Perceptron (MLP)**

The Multi-Layer Perceptron usually has one or more hidden layers between its input and output layers. Its extended structure allows the MLP to successfully overcome the linear separability limitation of the single-layer perceptron and enable the network to learn more complex tasks with the supervised learning algorithm known as the error back-propagation algorithm. Its expressive power and well-defined training algorithms have made MLP a popular network architecture to be considered when using ANNs.

We applied a one hidden layer structure in MLP (a three-layer network). The non-linear activation function hyperbolic tangent sigmoid is chosen for the neurons in the hidden layer of our MLP [27]:

\[ \varphi(x) = \tanh \alpha x = \frac{\sinh \alpha x}{\cosh \alpha x} = \frac{e^{\alpha x} - e^{-\alpha x}}{e^{\alpha x} + e^{-\alpha x}} \] (3)

where \( \alpha \) is slope parameter and \( x \) is the inputs. The hyperbolic tangent sigmoid activation function transforms the net inputs to saturate an output class between -1 and +1. Figure 7 presents the hyperbolic tangent sigmoid function.

![Hyperbolic tangent sigmoid activation function](image)

**Figure 7. Hyperbolic tangent sigmoid activation function**

**Radial Basis Function (RBF)**

Radial Basis function (RBF) perceptron is another popular feedforward neural network topology alternative to the MLP. RBF generates outputs based on the distance between the inputs and the centroid of the neurons. RBF usually has one hidden layer where the activation function of each neuron is a radial function. In application, the distance usually takes the form of Euclidean distance and radial function usually takes the Gaussian form [27]:

\[ \varphi(x) = \exp(-\frac{1}{2\sigma^2} \|x - c\|^2) \] (4)
where \( x \) is the inputs and \( c \) is the centroids of the neurons. Figure 8 shows the Gaussian function model.

![Gaussian function model](image)

**Figure 8. Gaussian function**

During the training, the RBF starts with one initial neuron and keep adding neurons to the network until the input space is covered and the mean-square error of the predicted output and desired output reaches the goal [11].
3. EXPERIMENTS & RESULTS

3.1. Wound Image Collection

The wound images applied in our project are drawn from cases of human subject study carried out at Drexel University from 2006 to 2008 with obtained approval [28]. We recruited wound images from nineteen patients. All the images are photographs taken by commercial camera (FujiFilms® FinePix S700 digital camera) with cross-polarized filters in front of the flash and the lens, which reduced light reflection from the wound surface and provided a better condition for the later processing.

The resolution of each wound image is $3,072 \times 2,304$, and all images were saved in JPEG format. Each patient has 2 to 10 traced digital color image records of the wound healing process. We ruled out five from nineteen patients whose wound regions are hard to see or already cured, leaving fourteen patients with total of ninety-two available wound images, which will be used for training and testing the ANNs in the identification system.

In order to assess a predictive model for accuracy estimation, collected data are usually divided into different subsets. Model is built on one subset (called the training dataset), and validity of the model is tested on the other subset (called the validation dataset or testing dataset) [29]. For this purpose, "holdout validation" is the most intuitive method, where less than a third of the original data are randomly
chosen as testing dataset, which is mutually exclusive of the remaining data that form training dataset. However, if the original data are not large enough to be split at a ratio of 1:2 and keep representative of the problem as well, another strategy called “k-fold cross validation” is often used. In this method, the dataset is divided into k small subsets, where each subset acts as an independent holdout testing dataset once for the model trained with the rest of k-1 subsets in the repeated k times (the folds) validation process [30]. Five and ten are often the moderate k values to be considered.

Considering the representative distribution of collected wound images with different types as well as the computational expensiveness, we combine the features of the two method mentioned above and decide to prepare our training and testing database in the way of holdout at a 5-fold cross validation ratio. Moreover, an internal validation procedure applied in the design of ANNs (which is detailed in the later section) further ensures the validity and generalization of our identification system.

Hence, according to the split ratio of 1:4, one image is picked randomly from images of each patient to form a testing database, which consists of fourteen wound images with very good representativeness for validation, and the rest seventy-eight images are grouped as a training database for the system to be trained.

3.2. Training Data Preparation

The wound region in the images from both the training database and the testing database are manually traced through our graphical user interface (GUI) using the “poly” tracing tool to better adapt to the irregular shapes and vague boundaries of
different types of wounds. One of the manually traced images from the training database is illustrated as an example in Figure 9.

![GUI for manual trace of the wound region](image)

**Figure 9.** GUI for manual trace of the wound region

In order to automatically determine the best parameters of the four methods in our integrated segmentation approach, we classified the fourteen patients into four groups for the four different image segmentations, according to their general wound type which has similar image features. Based on the overlap scores of each group of images, two parameters in K-means clustering, four parameters in edge detection, one parameter in thresholding and two parameters in region growing are automatically fine-tuned and the optima are determined. The optimized parameters saved as default values are shown in the Table 1.

Before applying the segmentation, all images are rescaled with a “scale” value of -300, which is listed as default value for “scale” in the Table 1. This results in the
adaptive contraction of each image to 225 x 300 pixels from the original 3,072 x 2,304. Segmentation is then implemented on the rescaled image with the optimized parameters.

After the segmentation with the optimized parameters, the resulting polygons are first rescaled back to match the resolution of the original image, and then pass through the filter. The default value of parameter “minarea” for the filter is 0.4x10^{-3}, as listed in Table 1, indicating that the segmented polygons that occupy less than 0.04% of the whole image will be eliminated from the candidate polygons to be considered as target wound region. Another parameter for the filter is “maxarea” with default being 0.99, meaning if a segmented polygon is larger than 99% of the whole image will be filtered. This filtering procedure greatly reduces the quantity of the candidate polygons for the training, and resulted in a total of 1,451 polygons from the seventy-eight images in the training database.

Forty-nine features, as described in Table 3, are extracted to form a representative feature vector for each segmented and filtered polygon. Their corresponding overlap score is also calculated based on the MCC algorithm. The feature vectors are then fed into the system as the inputs and the overlap scores are used as the desired outputs for the training procedure of the ANNs.

Take one wound image from the training database as an example, a total of sixteen filtered polygons are generated by the four image segmentation methods for that image. As a demonstration, we pick three typical polygons out of the sixteen polygons shown in Figure 10 with their corresponding extracted feature vectors and
overlap scores listed in Table 4. The three segmented polygons are the results from the segmentation method, K-means clustering, edge detection, and thresholding, respectively.

![Example of three picked segmented polygons by method of: a) K-means clustering; b) Edge detection; c) Thresholding](image)

**Figure 10.** Example of three picked segmented polygons by method of: a) K-means clustering; b) Edge detection; c) Thresholding

**Table 4.** Example of corresponding feature vectors and overlap scores (MCC) of the three picked segmented polygons shown in Figure 7

<table>
<thead>
<tr>
<th>feature vector (total of 50 features)</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>a 1164 1789 ... 0.001581 0.796568 650.5 1091.5 0 1</td>
<td>0.9463</td>
</tr>
<tr>
<td>b 1154 1850 ... 0.0019 0.792229 609.5 1050.5 0.0039 1</td>
<td>0.8269</td>
</tr>
<tr>
<td>c 1349 1973 ... 5.05E-05 0.858403 805.5 1428.5 0 0.8705</td>
<td>-0.012</td>
</tr>
</tbody>
</table>
3.3. Designing the Neural Networks

For the identification system, we applied the MLP and RBF ANNs mentioned in the previous section, and implement them in the environment of MATLAB R2012a with Neural Network Toolbox™ 7 [27].

In our three-layer MLP, the MATLAB function `feedforwardnet` is used as a basis to design the network. Feature vectors as training inputs will be fed into the network which will get trained by a backpropagation training function called Levenberg-Marquardt optimization, and updates the connection weight and bias values iteratively until the Mean Squared Error (MSE) between the MLP’s prediction outputs and the desired outputs (overlap scores) is minimized.

A number of five is chosen for the hidden neurons to form the hidden layer. A smaller number does not give a stable prediction while a larger one always encounters the overfitting problem and gives poor performance. The final structure of the designed MLP is shown in Figure 12 (a).

In RBF, the number of neurons and the width of the radial function (represented by the parameter variable “MN” and “spread” in MATLAB function `newrb`) are critical to the design of the network. However, there is no clear rule of thumb for selection of these parameters, and sampling is not a feasible approach when the data for processing is large. In order to optimize the performance of the RBF algorithm, we designed a cross-validation procedure to decide the network’s parameters.
In the validation procedure, 20 percent of the training feature vectors with their corresponding overlap scores are randomly picked as a validation dataset. The dataset is used to train RBF with different combination of the parameters: the parameter variable “spread” is increased within a large range, for example from 1 to 300; and for each “spread” value, an incremental “MN” from 1 to 300 is applied.

MSE is used as the criteria for success. The MSE values are recorded over “MN”s for each “spread” value. An example is shown in Figure 11 (a) with records for five certain “spread” values; the minimum MSE value is also recorded for each “spread”, which can be seen in Figure 11 (b) as an illustration.

The validation procedure record (Figure 11) indicates that the optimized parameters for the RBF are a hidden neuron number of 40 and a “spread” value of 230. The final structure of the designed RBF is shown in Figure 13 (a).
Figure 11. (a) MSE values over neuron numbers for different “spread” value. 40 as a number of neurons is where all the MSE values have a significant fall; (b) MSE values over “spread” values. Global minimum is achieved at “spread” of 230
3.4. Training Results

After construction of the MLP and RBF ANNs, the 1,451 feature vectors and their corresponding overlap scores are fed into the networks for training. The training performance of MLP is shown in Figure 12 (b). This procedure requires approximately 13 seconds.

Figure 12. (a) The structure of the designed MLP, (b) Training performance of the designed MLP
The training performance of RBF is shown Figure 13 (b), and it takes about 2 seconds to get the network trained.

Figure 13. (a) The structure of the designed RBF, (b) Training performance of the designed RBF
3.5. Testing Results

After the Neural Networks applied for the identification system are fully trained, images from the testing database are used to test and compare the efficiency of the system with the two types of ANNs respectively.

Each of the fourteen testing wound images is put through the procedure of rescaling, image segmentation, filtering, and feature extraction, resulting in fourteen representative feature vectors sets to be fed into the identification procedure for the prediction results.

In order to evaluate the prediction results by our identification system, the testing images are all manually traced to provide overlap scores for the segmented and filtered polygons of each image. Using the overlap scores as well as the visualization of the segmented and filtered polygons, we are able to choose polygons to be the desired polygons, which should be considered and identified by the system for each image.

As each image is processed by the four different segmentation methods, the wound region might be segmented out by more than one segmentation method; therefore, it might have multiple desired polygons. For example, the visualized segmented and filtered polygons of a particular testing image are shown in Figure 14, and the desired polygons are the second (form K-means clustering), the tenth (form thresholding) and the thirteenth (form region growing).
The feature vectors are then fed to the prediction system, which generates corresponding predicted overlap scores. The polygon with the highest score is the predicted wound region for each image. If the predicted polygon is the same as the desired polygon, the wound region of that image is considered to be successfully identified by the prediction system. The statistical results of the testing from MLP and RBF are shown in the Table 5, where the prediction result is eleven out of fourteen images correct for MLP, and twelve out of fourteen images correct for RBF.

Figure 14. The visualization results of segmented and filtered polygons by four segmentation methods on one test image. The desired polygon number is the second, tenth and the thirteenth (from left to right)
Table 5. Comparative performance of the two ANNs

<table>
<thead>
<tr>
<th></th>
<th>Time used for training (second)</th>
<th>Correct Prediction (%)</th>
<th>training MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP (5)*</td>
<td>12.6636</td>
<td>71.4 (10/14)</td>
<td>0.010681</td>
</tr>
<tr>
<td>RBF (40)**</td>
<td>1.7175</td>
<td>85.7 (12/14)</td>
<td>0.019551</td>
</tr>
</tbody>
</table>

* Number of neurons in the hidden layer of MLP,
** Number of neurons in the hidden layer of RBF.

In order to further analyze the efficacy of the system, we examine the ranking of the prediction score of the desired polygon. If the desired polygon is not the top ranked, but is ranking greater than most of the segmented and filtered polygons for a testing image, the prediction and the identification system can still be considered to be relatively effective for that image.

The evaluation and the comparison of the ranking results are shown in Table 6.
Table 6. Comparative prediction with their ranking evaluation of the two ANNs

<table>
<thead>
<tr>
<th>Test image number</th>
<th>Number of segmented polygons</th>
<th>Number of desired polygons</th>
<th>Ranking of desired polygons</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>MLP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>#1</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>1</td>
<td>1 N/A</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>1</td>
<td>2 N/A</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>2</td>
<td>1 3 N/A</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>1</td>
<td>1 N/A</td>
</tr>
<tr>
<td>6</td>
<td>21</td>
<td>2</td>
<td>1 2 N/A</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>2</td>
<td>2 4 N/A</td>
</tr>
<tr>
<td>8</td>
<td>19</td>
<td>1</td>
<td>1 N/A</td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td>2</td>
<td>1 3 N/A</td>
</tr>
<tr>
<td>10</td>
<td>18</td>
<td>2</td>
<td>1 2 N/A</td>
</tr>
<tr>
<td>11</td>
<td>17</td>
<td>1</td>
<td>1 N/A</td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td>3</td>
<td>2 3 4</td>
</tr>
<tr>
<td>13</td>
<td>21</td>
<td>1</td>
<td>3 N/A</td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>2</td>
<td>1 2 N/A</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>18</strong></td>
<td><strong>1.7</strong></td>
<td><strong>1.4</strong></td>
</tr>
</tbody>
</table>
4. DISCUSSION & CONCLUSION

We proposed and designed a system to automatically identify the wound region from clinical digital wound images. There is a one-time cost of training with a set of manually traced images, and once trained, the system is fully automatic.

In order to solve the issue of differing applicability of image segmentation methods to particular domains, four segmentation methods are packed into an integrated segmentation approach for each image. This proposed way is not complicated but proved very effective and easy to implement, and showed a good performance as a sufficient prerequisite results to support the following wound identification tasks.

Additionally, automatic optimization of segmentation parameters is an improvement over the traditional methods of trial experience and prior knowledge. Automatic optimization facilitates quantifiable evaluation of parameter choice for a group of images with similar features, and is very suitable for our task in particular as it guarantees a new wound image with distinct wound type can be well-segmented by at least one segmentation method with fine-tuned parameters from the integrated segmentation approach.

Preprocessing procedures such as image rescaling and polygon filtering improve the performance of the system. Our results show that these steps are significant factors in the efficiency of the system, and can make it possible for the
complete identification of a wound region in a novel image to cost less than one minute of processing time.

In the prediction process of the system, two types of ANNs are applied with a supervised learning strategy. The traditionally tolerated uncertainty of network parameters optimization is solved by the proposed cross-validation approach. With these automatically determined parameters, the designed networks produce satisfactory prediction results.

The prediction results and the comparison of the two ANNs detailed in Table 5 indicate that both regulated MLP and RBF have decent efficiency. The testing results also indicate that the two types of networks have different advantages and disadvantages.

MLP has better generalization capability, as we can observe from the stable, high ranking of the desired polygons in the testing procedure; although the exact accuracy rate of its prediction is less competitive with the RBF network in a testing database of fourteen images, which might make a difference in a much larger database. Another drawback of MLP is the training time for the network, especially when the training database is large. However, because the training procedure is a one-time cost process, this is generally a tolerable issue. As the performance criteria we applied are calculated based on the measure of MCC which outputs value ranging from -1 to 1, the Mean Squared Error (MSE) of both networks in the level of 0.01 are considered quite good, while the MLP has a slightly better performance than the RBF.
Given proper determination of the parameters for the network design, the advantages of RBF over MLP are obvious. The comparative statistic in Table 5 shows that the training time of RBF is significantly less than that of MLP, and its prediction accuracy is more competitive. Another drawback of MLP is that determination of its parameters follows no clear rules, and cross-validation approach that we proposed could take considerable time. However, this is acceptable as it is a one-time cost process. Although the prediction accuracy of RBF for the total fourteen images is better than MLP, we can see from Table 6 that its deviation of the prediction results is worse, and this is mainly due to its local nature of the approximation capability.

Improvement works could be considered to apply in the assessment of the trained system. For example, an actual k-fold cross validation might be taken instead of the holdout validation, to further warrant the validity and reliability of the system, by evaluating the average estimation of the k times validation results. In our project, wound images collected are all the photographs taken under the unified standard; and when we prepare for the database, some collected wound images are manually excluded on account of biological motivation. For the final application in the clinical practice, original wound images with natural variation or changed characteristic could be included to validate the sensitivity and robustness of the system. Each method applied in the system, such as the image segmentation methods, could be then further refined or replaced by more sophisticated one accordingly. The strategy of re-training the ANNs by including novel and distinct samples encountered in the practice could also be a solution for the robustness of the system.
In summary, the satisfactory results of the identification system demonstrate that the proposed methods and implemented system are a promising method for fully automated clinical wound assessment, and the reduction of human error and ambiguity in wound categorization guarantees the greater accuracy. Future works may include further optimization of the system based on these preliminary results.
LIST OF REFERENCES


