AUTOMATIC SEGMENTATION OF PSORIASIS IMAGES USING NEURO-FUZZY TECHNIQUES

J.S. Taur
Department of Electrical Engineering,
National Chung Hsing University, Taichung, Taiwan.

C.W. Tao
Department of Electrical Engineering,
National I-Lan Institute of Technology, I-Lan, Taiwan.

C.C. Chen **and** *C.W. Yang*
Veterans General Hospital, Taichung, Taiwan.

ABSTRACT

In this paper, we develop an automatic method for psoriasis image segmentation using neuro-fuzzy techniques, which can be used in an therapy evaluation system. Since the psoriasis is a chronic disease, it is important to track the condition of the patient to select proper treatment. The psoriasis images are segmented into normal skin regions and abnormal regions automatically. The areas of each kind of regions of a patient at different moments can then be estimated. This information can be used to give a quantitative measure of the progress of the treatment. The provided information can avoid the variation of the human factor in the evaluation procedure and can offer a objective index for the doctor to select the most suitable treatment for the patient. For comparison, we also implement a 2D color clustering algorithm which was used to segment digitized dermatoscopic images. In the experiments, our proposed approach provides better performance.

1. INTRODUCTION

Psoriasis is a chronic disease and has the special characteristics of scaly papules and plaques[6]. The origin of the disease is not clear and might be related to gene. Many persons have the potential to have psoriasis. Several factors, such as stress, may trigger the development of psoriasis. Environmental influences may modify the course, severity, and age of the individual at the time of the onset of the disease. The disease usually subsides and recurs repeatedly. Therefore, the patients are emotionally and physically disabled. The lesions of psoriasis are distinctive visually. The psoriasis regions are usually overlapped oval with deep red color. The scale is adherent, silvery white, and reveals bleeding points when removed.

There are several morphological variations of psoriasis. Among them, the chronic plaque psoriasis is a well-defined plaque and it is the most common presentation of psoriasis. A temporary brown, white, or red macula remains when the plaque subsides. A typical case is shown in Fig. 1. In this paper, we focus on the segmentation of the chronic plaque psoriasis images into the normal skin and the psoriasis regions using neuro-fuzzy techniques. For the evaluation of the treatment, the images are segmented and the psoriasis areas are measured before and after the treatment. The percentage of the decrease in the area of psoriasis will show the improvement index of the treatment.

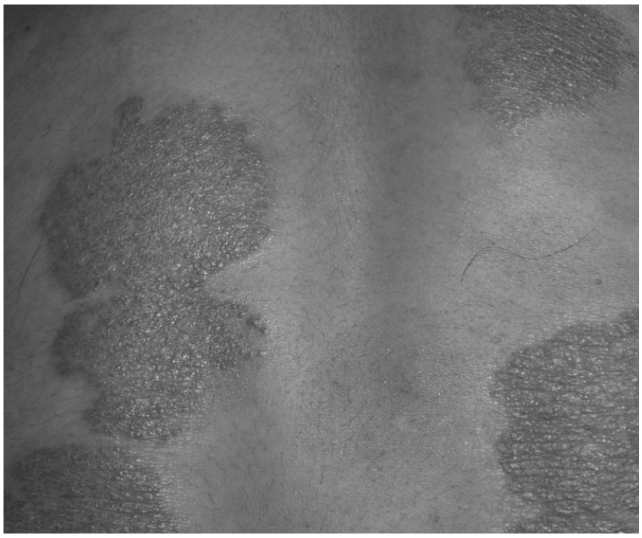

Figure 1: This figure shows a typical psoriasis image.

Since the skin color and the illuminance condition may be very different for each case. We can not use a single neuro-fuzzy network which is trained beforehand to segment all the images. In order to obtain accurate segmentation, the neuro-fuzzy network needs to be trained for each image to reflect the variations. The proper training regions can be selected for each image manually. However, as the number of images increases, this region selection process will become tedious. Therefore, it will be more convenient if the sys-

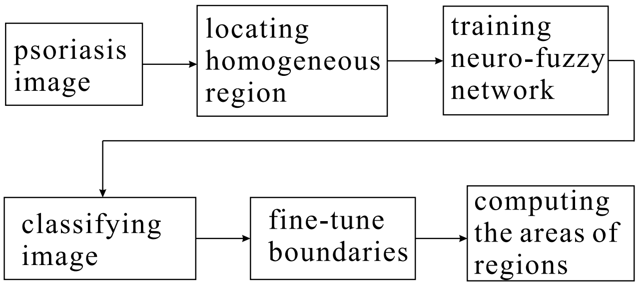

Figure 2: Block diagram of the automatic psoriasis image segmentation system.

tem can analyze the image and locate suitable training regions. In our approach, the homogeneous regions are first located. These regions are grouped into two classes, i.e. normal skin and psoriasis. Then feature vectors are extracted and used to train the neuro-fuzzy network. After that, the neuro-fuzzy network is used to segment the whole image. Small regions are then removed using morphological operations to obtain a smoothed results. The flow chart describing our approach is shown in Fig. 2.

The paper is organized as follows. In Section 2., the feature vector and the algorithm for the detection of training regions is described. Section 3. describes the neuro-fuzzy classifier adopted to segment the image. The experiment results and comparison study are given in Section 4.. Finally, Section 5. concludes the paper.

2. AUTOMATIC SELECTION OF TRAINING REGIONS

As mentioned in the previous section, the variation in skin color can be very large among the images. Therefore, each image has to be examined to obtain the training data for the neuro-fuzzy network. It is assumed that the homogeneous regions only contain the pixels from a single class, normal skin or psoriasis. Then the homogeneous regions can be used as training regions for the neuro-fuzzy network. In the following, we will describe the feature vector and the algorithm for the detection of homogeneous regions.

2.1 Feature Extraction

The feature vector is composed of two parts. The first part contains the texture information from gray level image. In our approach, the fuzzy texture spectrum [8], which has been successfully applied to texture classification, is adopted as the texture feature. It is based on the relative gray levels between pixels. A vector of fuzzy values will be used to indicate the relationship of the gray levels between the neighboring pixels. The fuzzy texture spectrum can be considered as the distribution of the fuzzified differences between the neighboring pixels. It is insensitive to the noise and the changing of the background brightness in texture images. When tested with twelve Brodatz texture images, the rate of classification error can be as low as 0.2083%. The second part is the statistical characteristics extracted from the hue-saturation plane. We project the points in the normalized RGB space onto the plane determined by the points with coordinates (0,0,1), (0,1,0), and (1,0,0)(cf. Fig. 3). On this plane, we can define a new two-dimensional coordinate system. Two-dimensional histogram is used to describe the distribution of the projected points. The feature extraction algorithm is depicted in Fig. 4.

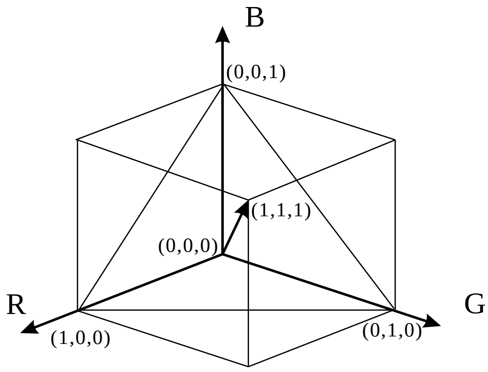

Figure 3: This figure shows that the points in the cube of the normalized RGB space can be projected to the plane defined by (0,0,1), (0,1,0), and (1,0,0) along the [1 1 1] direction.

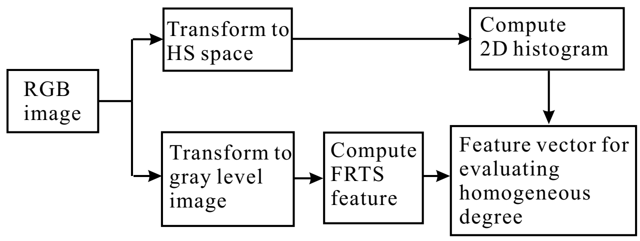

Figure 4: This figure shows the feature extraction algorithm for the detection of homogeneous regions.

2.2 Detection of Homogeneous Regions

In order to collect the training data for the neuro-fuzzy classifier, we develop a method to automatically locate the homogeneous regions that contain pixels from the same class. We assume that if the texture and the 2D histogram are homogeneous in a region, then the pixels in this region are of one single class. The moving windows centered on the coarser grid points defined on the original image are examined to see if they are homogeneous (cf. Fig. 5). The image block B covered by the window is divided into four small blocks in three different ways as in Fig. 6. For each decomposition, the feature vectors f_{TH}'s for the small blocks, including the fuzzy texture spectrum and the 2D histogram, are computed and the largest distance between the feature vectors is recorded. The largest distance d_L among the different decompositions is selected as a measure for the degree of inhomogeneity of the block B.

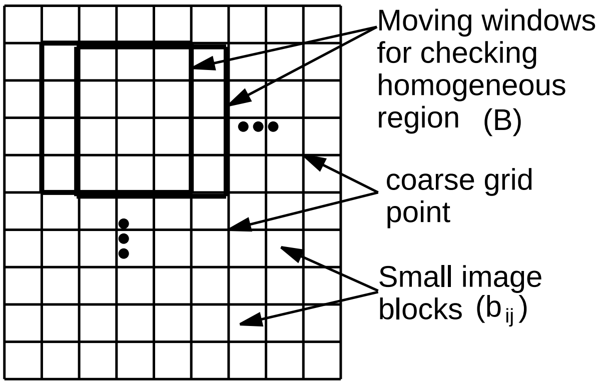

Figure 5: This figure depicts the diagram for locating homogeneous regions.

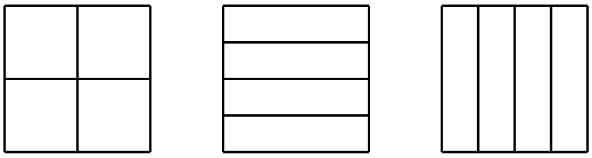

Figure 6: This figure shows the partition of the image block B from the moving window for checking homogeneous regions.

Each small image block b_{ij} (cf. Fig. 5) is associated with a element $M_h(i,j)$ of a matrix M_h. If the largest distance d_L of the block B is smaller than a threshold, the block is regarded as a candidate for a homogeneous block and the corresponding elements in M_h are increased by one. Therefore, the element $M_h(i,j)$ summarizes the count of the moving windows which considers the block $b_{i,j}$ as homogeneous. This process is performed on each grid points. Then the small block b_{ij} with $M_h(i,j)$ larger than a threshold is marked as a homogeneous region. In our design, if 75% of the moving windows which cover the small block are homogeneous, the small block is considered as a homogeneous region.

Selection of the Threshold for d_L. The threshold can be determined by examine the distribution of the difference d_L. We found that the d_L's from the moving windows that cover normal skin and psoriasis are relatively large. The d_L's in the normal skin regions are usually small. And the d_L's in the psoriasis regions fall between the values from the other two situations. Therefore, the distribution of d_L from the whole image is modeled as a Gaussian mixture of three components. EM algorithm is adopted to compute the parameters in the model. Then the threshold is set to be the intersection point of the two Gaussian components with larger mean values.

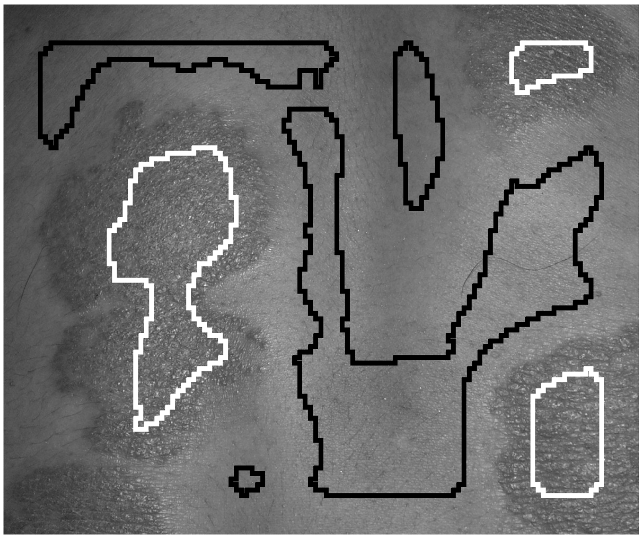

Figure 7: This figure shows a typical result for the homogeneous region detection.

2.3 Classification of Regions

After thresholding the image using the distance values, there may exist many uniform regions. In order to extract training data from these regions, they have to be classified into normal skin or psoriasis regions. The feature f_{TH} is computed for each region. The regions are merged according to the distance between f_{TH}'s. The regions with the smallest distance between f_{TH}'s are merged. The merging process continues until the number of regions is two. A typical result after the classification procedure is shown in Fig. 7. The regions with blue and yellow borders are regions of normal skin and psoriasis, respectively.

3. NEURO-FUZZY CLASSIFIER

After the homogeneous regions are identified, the features from these regions are extracted and used to train the neuro-fuzzy network. The feature vector used for the classification of each pixel is the same as that for the detection of homogeneous region except that the size of the image block used for feature extraction is smaller. That is, only the small neighborhood of a pixel is used to classify the pixel. In the following, we describe the classifier we adopt.

Several neural and neuro-fuzzy networks have been proposed to solve classification problems[1, 4, 5]. In the design of a multi-layered perceptron neural network, usually the parameters in the network have no physical meanings and their initial values are randomly selected. On the other hand, the parameters of neuro-fuzzy classifiers have clear physical meanings and it is easier to design the initial values from the input- output data pairs. Moreover, the neuro-fuzzy systems provide an effective framework to incorporate human linguistic descriptions into the classifiers[11]. In this paper, we propose a new neuro-fuzzy network for classification.

When the neuro-fuzzy network is designed by using the input-output data pairs $(\mathbf{x}^i, f^i)$ from experiments, a rule "If $\mathbf{x}$ is $\mathbf{x}^i$, then f is f^i" can be constructed from each data pair for simplicity. However, when the number of samples is large, the computational cost will make this simple approach inadmissible. In this situation, a clustering algorithm can be applied to find the "representative" centroids of the data and then assign importance factors to the corresponding rules to reflect the numbers of samples covered by the centroids. From this idea, we propose a neuro-fuzzy classifier (NEFCAR) with adjustable rule importance which provides both positive and negative rules[9, 10]. That is, the connections between the rule nodes and the output nodes can be either +1 or -1 and each connection is associated with a factor to reflect the importance of the rule in the defuzzification stage. The defuzzification method is similar to that proposed in [2]. We test the performance of the classifier using the iris data and the on-line face detection and recognition application. Good classification results are obtained in both cases[9, 10].

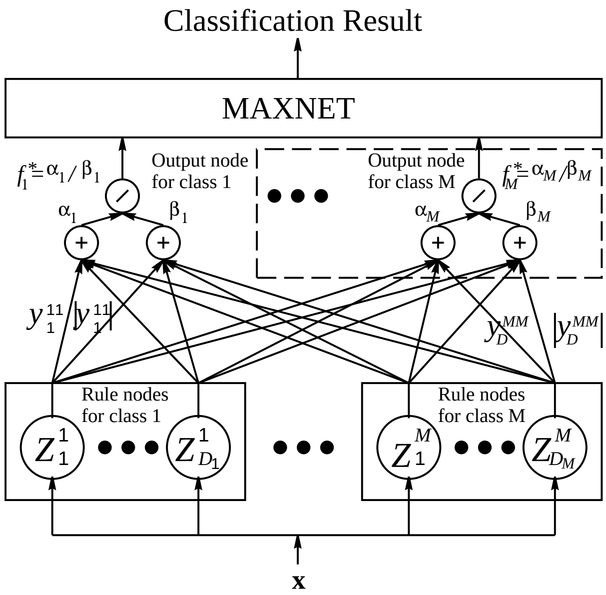

Figure 8: Block diagram of the neuro-fuzzy classifier.

3.1 Architecture of NEFCAR

To facilitate the discussion, we assume that the input pattern, the number of input patterns, the dimension of input pattern, the number of class, and the number of rules in class j, are denoted as $\mathbf{x}$, N_x, N, M, and D_j, respectively. The fuzzy rules of NEFCAR for an N-dimensional input pattern $\mathbf{x}=\{x_1, \ldots, x_N\}$ are of the form:

$$\begin{cases} \text{Positive Rule :} & \text{If } x_1 \text{ is } \mu_1 \text{ and } \cdots \text{ and } x_N \text{ is } \mu_N, \\ & \quad \text{then } \mathbf{x} \text{ is in class } i \\ \text{Negative Rule :} & \text{If } x_1 \text{ is } \mu_1 \text{ and } \cdots \text{ and } x_N \text{ is } \mu_N, \\ & \quad \text{then } \mathbf{x} \text{ is } \textit{not} \text{ in class } k \end{cases} \tag{1}$$

where μ_l's are the fuzzy sets with Gaussian-like membership functions. The consequent of the positive rule, "$\mathbf{x}$ is in class i", is represented as a classifier output variable $f_i = 1$. On the other hand, the consequent of the negative rules is represented as an output $f_k = -1$. The schematic diagram of the architecture of the classifier is shown in Fig. 8, which is designed for a M-class classifier. In our design, each class has its own set of rule nodes. And rule nodes are fully connected to the output nodes. A rule acts as a positive rule when it contributes to the output of its own class. Otherwise, it acts as a negative rule. That is, if a rule node is from class j and the output of class j is under consideration, then this rule is regarded as a positive rule. On the other hand, if the output of class $i(\neq j)$ is under consideration, this rule is designed to behave as a negative rule. Therefore, for a rule node in class j, we have effectively one positive rule and $(M-1)$ negative rules as follows.

$$\begin{cases} \text{Positive Rule :} & \text{If } x_1 \text{ is } \mu_1 \text{ and } \cdots \text{ and } x_N \text{ is } \mu_N, \\ & \quad \text{then } f_j = 1 \\ \text{Negative Rule :} & \text{If } x_1 \text{ is } \mu_1 \text{ and } \cdots \text{ and } x_N \text{ is } \mu_N, \\ & \quad \text{then } f_1 = -1 \\ & \vdots \\ \text{Negative Rule :} & \text{If } x_1 \text{ is } \mu_1 \text{ and } \cdots \text{ and } x_N \text{ is } \mu_N, \\ & \quad \text{then } f_{j-1} = -1 \\ \text{Negative Rule :} & \text{If } x_1 \text{ is } \mu_1 \text{ and } \cdots \text{ and } x_N \text{ is } \mu_N, \\ & \quad \text{then } f_{j+1} = -1 \\ & \vdots \\ \text{Negative Rule :} & \text{If } x_1 \text{ is } \mu_1 \text{ and } \cdots \text{ and } x_N \text{ is } \mu_N, \\ & \quad \text{then } f_M = -1. \end{cases} \tag{2}$$

The importance of the d_{th} rule in class j to the output of class i is represented by a weighting factor y_d^{ij}. The importance factor is interpreted as duplicating the same rule y_d^{ij} times in the defuzzification procedure for output i. When y_d^{ij} gets larger, the corresponding rule becomes more important to output f_i^*.

Assume that the mean and the variance of the l_{th} (Gaussian-like) membership function of the d_{th} rule for the j_{th} class are denoted as m_{dl}^j and σ_{dl}^j, respectively. Moreover, the multiplication is adopted for the "and" operation in the rule in Eq. 2. Then the output of class i can be computed using the following formula.

$$f_i^* = \alpha_i / \beta_i \tag{3}$$

where

$$\alpha_i = \sum_{d=1}^{D_i} (+1) y_d^{ii} Z_d^i + \sum_{j=1, j\neq i}^{M} \sum_{d=1}^{D_j} (-1) y_d^{ij} Z_d^j \tag{4}$$

$$= \sum_{d=1}^{D_i} y_d^{ii} Z_d^i - \sum_{j=1, j\neq i}^{M} \sum_{d=1}^{D_j} y_d^{ij} Z_d^j \tag{5}$$

$$\beta_i = \sum_{j=1}^{M}\sum_{d=1}^{D_j} y_d^{ij} Z_d^j \quad (6)$$

and

$$Z_d^j = exp(-\sum_{l=1}^{N} \frac{(x_l - m_{dl}^j)^2}{(\sigma_{dl}^j)^2}) \quad (7)$$

is the matching degree of the if-part of the d_{th} rule in class j.

We can further absorb the (+1) and (-1) into y_d^{ij} and obtain

$$\alpha_i = \sum_{j=1}^{M}\sum_{d=1}^{D_j} y_d^{ij} Z_d^j \quad (8)$$

$$\beta_i = \sum_{j=1}^{M}\sum_{d=1}^{D_j} |y_d^{ij}| Z_d^j. \quad (9)$$

Therefore, we can consider (y_d^{ij})'s as weighting parameters in the network shown in Fig. 8. From the above equations, it is obvious that

$$-1 \leq f_i^* \leq 1 \text{ , for } i = 1, \cdots, M.$$

The classification procedure is described as follows. The pattern $\mathbf{x}$ is input to each rule node which computes the matching degree (Z_d^j) of the if-part of the rule. Then the output of each class (f_i^*) is computed according to the weighting of each rule. The class, c, which has the largest output is identified. That is,

$$c = arg(\max_i \ f_i^*). \quad (10)$$

The pattern is classified into class c. Similarly, if we are dealing with a two-class application, the shaded region and the weights connected to it, which are shown in Fig. 8, can be removed. In this situation, the sample is classified into class 1 if $f_1^* > 0$. Otherwise, it is classified into class 2.

3.2 Training Algorithm

The locally unsupervised and globally supervised technique is adopted. That is, for the parameters in the network of a given class, we utilize a unsupervised clustering method to obtain the initial values using the patterns in the same class. And training patterns from all classes are used to update the classifier after the initialization. The decision-based and approximation-based strategies are combined to provide a suitable amount of training for each training pattern in the supervised training. Therefore, the classifier can reach convergence quickly. The reinforced and anti-reinforced learning rules are adopted to training the neuro-fuzzy networks. The reinforced and anti-reinforced learning rules are given with different weighting so that the training can be more efficient. In the following, the training algorithm will be described.

3.2.1 Training Strategy

Initialization. For many classification applications, the linguistic rules for describing how to classify the data are not available. In this case, the initial conditions for the parameters in the neuro-fuzzy classifiers have to be derived from the data set. Clustering techniques (unsupervised training) can be applied to the data to derive fuzzy sets and initial rules from the fuzzy clusters. In this study, we do not have the prior linguistic classification rules. And we adopt the K-mean algorithm to cluster the training data to obtain the initial $(m_{dl}^j)'s$. The $(\sigma_{dl}^j)'s$ are set to one initially. The initial values for rule importance factors $(y_d^{ij})'s$ are set to one when $i = j$. Otherwise, $y_d^{ij} = -\delta$, where δ is small positive number (e.g. 0.1). That is, the importance factors for the positive rules are set to one and those for the negative rules are set to small negative numbers. More complicated algorithm can also be adopted to obtain the initial values [3].

Combined training strategy. In decision-based training, the teacher only indicates to which class the current pattern should belong. There are no target values for the training patterns. Only the misclassified patterns are used to train the system through the reinforced and anti-reinforced rules. On the other hand, in the approximation-based approach, the squared errors between output values and the teacher values for all the training data are taken into consideration. For training NEFCAR's, we adopt a strategy which combines the approximation-based and decision-based training strategies.

In our training approach, each pattern is assigned a target value (-1 or +1) for each output. Let the teacher vector for a pattern from class k be denoted as $T_k = \{t_1, t_2, \cdots, t_M\}$ and have the value

$$\begin{cases} t_i = -1, & i \neq k \\ t_i = +1, & i = k. \end{cases} \quad (11)$$

Similar to the decision-based training, not all the training patterns are involved in the updating. In our design, if the output reach certain noise tolerance value, the training is omitted. If the absolute value of $e_i = t_i - f_i$ is smaller than $(1 - n_t)$, the training pattern is considered to be safe and no training is needed. Note that from the formula for computing the output values in Eqs. 3–7, the output values always fall into the closed interval [-1,1]. Therefore, we do not need to normalize the output and the noise tolerance value can be easily selected. In this case, the pattern with larger error will be trained with a larger training ratio. This can give each data a proper training weighting without forcing all of them to produce the (artificially given) limiting values (-1 or +1). Therefore, the training procedure can reach a compromise between training patterns more easily and can converge faster.

When the output gets closer to the target value (-1 or +1), it will have better noise tolerant property. That is, the noise tolerance can be increased if we increase n_t. In this study, n_t is increased gradually in the training process. After all the training pattern are considered to be safe for certain noise tolerance level, the n_t is increased and the training continues.

3.2.2 Updating Formula

The goal of training is to minimize the mean squared difference between the output f_i^* and the teacher value t_i for the patterns with error that is above a noise tolerance level (1-n_t). That is, when the absolute value of $e_i = t_i - f_i^*$ is smaller than (1-n_t), the current training pattern is considered as a "safe" pattern, and the contribution to the training is ignored. Otherwise, the contribution to the training of the pattern at the i_{th} class output can be calculated using the gradient. Since Gaussian-like functions are adopted as the membership functions, the back-propagation algorithm can be easily implemented to calculate the gradients for all the parameters in the neuro-fuzzy classifier. In our design, the parameters in the classifier are updated after each training pattern is presented to the classifier. The gradient from the i_{th} output node can be computed as

$$\begin{cases} g_{ij} = 0, & \text{abs}(e_i) < 1 - n_t \\ g_{ij} = \frac{\partial e_i^2}{\partial w_j}, & \text{abs}(e_i) \geq 1 - n_t, \end{cases} \tag{12}$$

where w_j represents one of the parameters in the set $\{y_d^{ij}, m_{dl}^j, \sigma_{dl}^j\}$. The formula for a two-class (single output) situation are shown in the following. The updating formula for a multi-class situation can be derived similarly.

$$\frac{\partial e_1^2}{\partial w_j} = 2(f_1^* - t_1)\frac{\partial f_1^*}{\partial w_j}$$

$$\frac{\partial f_1^*}{\partial y_d^{1j}} = Z_d^j \beta_1^{-1}(1 - f_1^* \frac{y_d^{1j}}{|y_d^{1j}|})$$

$$\frac{\partial f_1^*}{\partial m_{dl}^j} = 2Z_d^j \beta_1^{-1}(x_l - m_{dl}^j)(y_d^{1j} - f_1^*|y_d^{1j}|)/(\sigma_{dl}^j)^2$$

$$\frac{\partial f_1^*}{\partial \sigma_{dl}^j} = 2Z_d^j \beta_1^{-1}(x_l - m_{dl}^j)^2(y_d^{1j} - f_1^*|y_d^{1j}|)/(\sigma_{dl}^j)^3$$

Assume that the current training pattern is from class k. Furthermore, let i and j denote the indexes of the output node and the rule node respectively. Then the parameter can be adjusted by the updating rule:

$$w_j^{p+1} = w_j^p - \sum_{i=1}^{M} g_{ij}\ \gamma(i,j,k) \tag{13}$$

where p is the iteration index and γ is a weighting factor which will be explained below.

For the architecture in Fig. 8, we can compute three types of training gradient information for a pattern from class k according to which output node is under consideration and to which class the rule belongs:

$$\begin{array}{lll} \text{1. reinforced learning:} & & \\ \quad g_{ij},\ j = k & \gamma = \eta_1 & \\ \text{2. anti-reinforced learning:} & & \\ \quad g_{ij},\ i = k,\ \text{and } j \neq k, & \gamma = \eta_2 & (14) \\ \quad g_{ij},\ i \neq k,\ j \neq k,\ \text{and } i = j, & \gamma = \eta_2 & \\ \text{3. cross-reinforced learning:} & & \\ \quad g_{ij},\ i \neq k,\ j \neq k,\ \text{and } i \neq j, & \gamma = \eta_3 & \end{array}$$

where $g_{ij} = \partial e_i^2 / \partial w_j$ and w_j is a parameter related to rule node j (cf. Eq. 12.) In the first situation, the updating will try to increase the absolute value of the outputs of the rule nodes and the absolute values of the importance weighting factors for the rules from the same class as training pattern class k to decrease the error. We call it "reinforced learning". In the second case, the updating will try to decrease the absolute value of the outputs of the rule nodes and the absolute values of the importance weighting factors for the rules from classes other than k. This type of training is called anti-reinforced learning. In the last case,the training will try to increase the outputs of the rule nodes and the absolute values of the importance weighting factors and is called "cross-reinforced learning".

Although the cross-reinforced learning can also help the network reduce the training error, it may result in degrading the generalization performance. In Eq. 14, different types of training are assigned different weighting γ to avoid degrading the generalization capability. Let the weighting factors for the first, second, and third type be denoted as η_1, η_2, and η_3, respectively. Therefore, in our design,

$$\eta_3 \leq \eta_2 \leq \eta_1.$$

And η_3 is set to zero sometimes.

4. EXPERIMENT RESULTS

The procedure described in Fig. 2 is applied to several psoriasis images. Small regions in the segmentation results are removed using morphological operations to obtain a smoothed results. The segmentation results are shown in Figs. 9-11. The sizes of the images are 1000×1200, 1600×1000, and 1600×1000, respectively. We can observe that the proposed algorithm can produce good segmentation. For comparison, the method proposed in [7] is implemented. In their approach, the RGB image is transformed to the $L^*u^*v^*$ color space. A two-dimensional histogram is computed with the two principal components to select the number of valid clusters. The image is then segmented using a modified version of the fuzzy c-means (FCM) clustering technique. (In our experiment, the clusters are manually classified into two classes based on the segmented image.) Finally, the segmented image is cleaned using mathematical morphology, the region borders are smoothed, and small components are removed. One of the results is

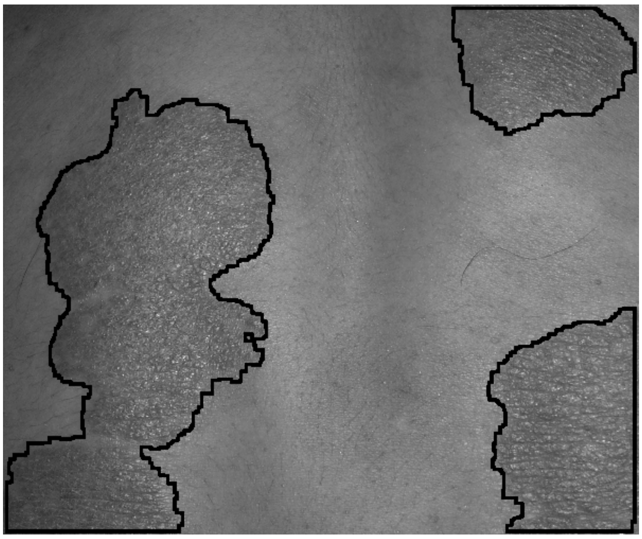

Figure 9: This figure shows segmentation results of psoriasis images.

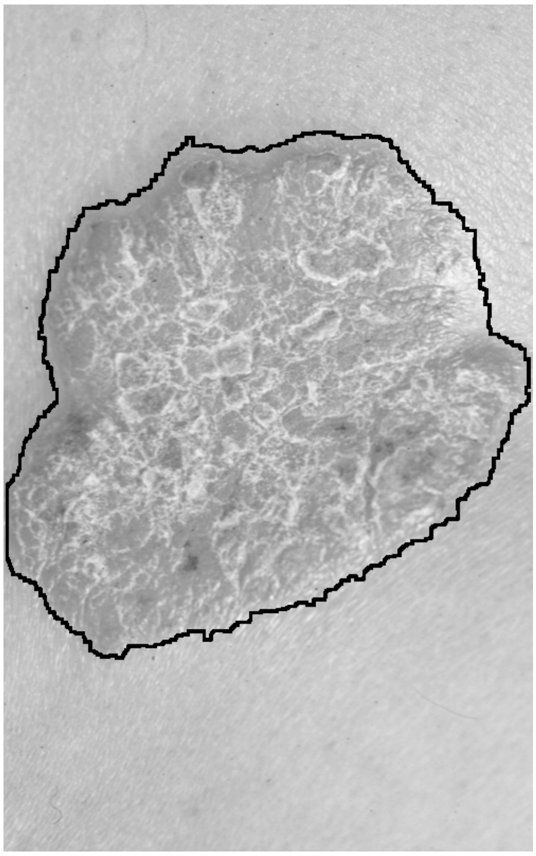

Figure 10: This figure shows segmentation results of psoriasis images.

shown in Fig. 12. In [7], the texture information is not considered and only the color information is adopted. Sometimes, the color of the scale in the psoriasis region will be very similar to that of the normal skin region due to the reflection. In this case, these regions cannot be distinguished if the classification is based on the color of the pixel only. Therefore, the result is not as robust as that by our approach.

5. CONCLUSION

In this paper, we develop an automatic method for psoriasis image segmentation using neuro-fuzzy techniques, which can be used in an evaluation system for the treatment of psoriasis. Each image is analyzed to locate the homogeneous regions for training data. Therefore, the variation between images will not affect the accuracy of the segmentation. Since the texture and color information are both taken into consideration, the segmentation is more robust than that produced by the method using only color information.

6. REFERENCES

[1] S. Abe and R. Thawonmas "A Fuzzy Classifier with Ellipsoidal Regions" *IEEE Trans. on Fuzzy System*, VOL. 5, NO. 3, August 1997.

[2] D. Driankov and H. Hellendoorn and M. Reinfrank, *An Introduction to Fuzzy Control*, Springer-Verlag, 1993.

[3] I. Gath and A.B. Geva, "Unsupervised Optimal Fuzzy Clustering", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 773-781, July, 1989.

[4] S.Y. Kung and J.S. Taur, "Decision-based Neural Networks with Signal/Image Classification Applications," *IEEE Trans. on Neural Networks*, Vol. 6, No. 1, pp. 170-181, January, 1995.

[5] S.H. Lin, S.Y. Kung, and L.J. Lin "Face Recognition/Detection by Probabilistic Decision-Based Neural Network" *IEEE Trans. on Neural Networks*, Vol. 8, No. 1, pp. 114-132, January 1997.

[6] Thomas P. Habif, "Clinical Dermatology", The C.V. Mosby Company, second edition, 1990.

[7] P. Schmid, "Segmentation of Digitized Dermatoscopic Images by Two-Dimensional Color Clustering", *IEEE Trans. on Medical Imaging*, Vol. 18, No. 2, February 1999.

[8] J.S. Taur and C.W. Tao, "Texture Classification Using a Fuzzy Texture Spectrum and Neural Networks", *Journal of Electronic Imaging*, Vol. 7, No. 1, pp. 29-35, January 1998.

[9] J.S. Taur and C.W. Tao, "Face Detection Using Neuro-Fuzzy Classifier", *In Proc., International Symposium on Multimedia Information Processing*, pp.309-314, December 1998.

[10] J.S. Taur and C.W. Tao, "An On-line Face Detection and Recognition System Using Neuro-Fuzzy Classifier", *In Proc., International Symposium on Multimedia Information Processing*, pp.297-302, December 1999.

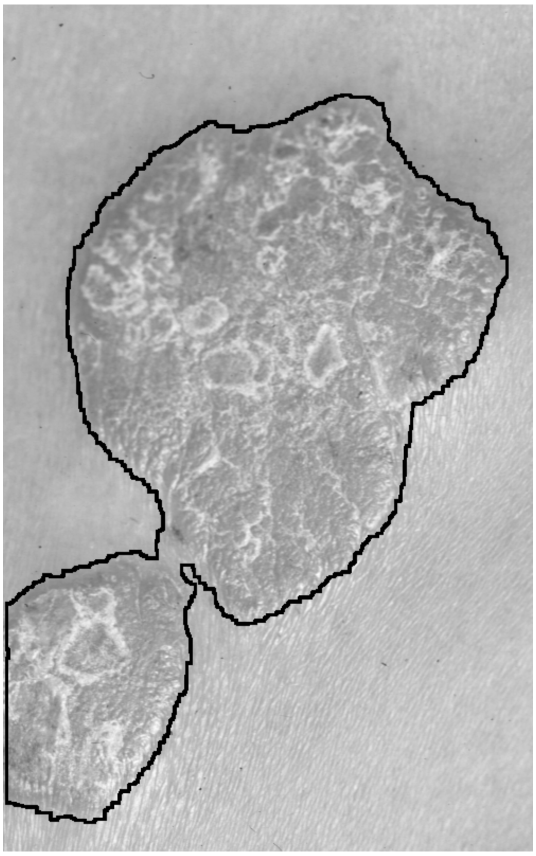

Figure 11: This figure shows segmentation results of psoriasis images.

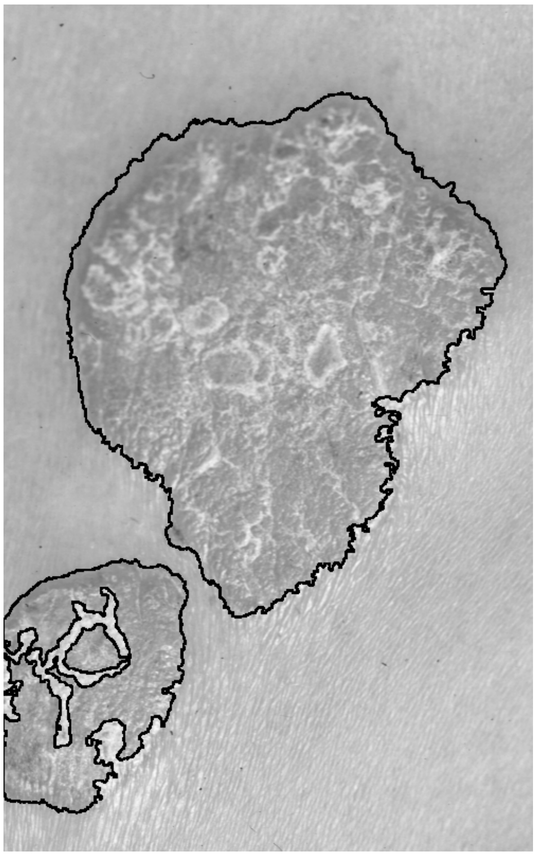

Figure 12: This figure shows a segmentation result using the method in [7].

[11] L.X. Wang, *Adaptive Fuzzy Systems and Control : Design and Stability Analysis*, Prentice Hall, 1995.