Segmentation of Breast Ultrasound Images Using Neural Networks

Ahmed A. Othman, Hamid R. Tizhoosh

HAL Id: hal-01571364
https://inria.hal.science/hal-01571364

Submitted on 2 Aug 2017

Segmentation of Breast Ultrasound Images Using Neural Networks

Ahmed A. Othman * and Hamid R. Tizhoosh **

Abstract. Medical image segmentation is considered a very important task for diagnostic and treatment-planning purposes. Accurate segmentation of medical images helps clinicians to clarify the type of the disease and facilitates the process of efficient treatment. In this paper, we propose two different approaches to segment breast ultrasound images using neural networks. In the first approach, we use scale invariant feature transform (SIFT) to calculate a set of descriptors for a set of points inside the image. These descriptors are used to train a supervised neural network. In the second approach, we use SIFT to detect a set of key points inside the image. Texture features are then extracted from a region around each point to train the network. This process is repeated multiple times to verify the generalization ability of the network. The average segmentation accuracy is calculated by comparing every segmented image with corresponding gold standard images marked by an expert.

1 Introduction

Most of the image thresholding methods deal with different images in a static way without paying any attention to the different properties of different images. These methods could generate good results for some images. A method of dynamic nature (with learning ability) that deals with the image based on its features is still needed. The features of the image can guide the process of calculating accurate thresholds for a given class of images sharing some common characteristics.

In this paper, we propose two different approaches to threshold medical images using neural network trained using a set of features extracted from the image. In the first approach, we use SIFT descriptors generated from a set of key points. These points seem to be the most descriptive points of the image and are used to train a supervised neural network that could be used to threshold new images. In the second approach, we use SIFT to detect a set of key points within the image. A rectangle around each point is generated and used to extract a set of second order texture features. These features are used to train a supervised neural network that could be used to threshold new images.

---

* Ahmed A. Othman is a Ph.D. candidate at the Department of Systems Design Engineering, University of Waterloo, Ontario, Canada, a4abdelr@uwaterloo.ca
** Hamid R. Tizhoosh is a faculty member at the Department of Systems Design Engineering, University of Waterloo, Ontario, Canada, tizhoosh@uwaterloo.ca

This paper is organized as: In sections 2 and 3 a brief survey of image thresholding and neural networks is provided. In section 4, the SIFT technique is explained. In section 5, the level set segmentation technique is reviewed since we use it for comparison purposes. In section 6, the proposed techniques are introduced and discussed. In section 7, medical images are used to test the proposed techniques. In section 8, the paper is summarized.

2   Image Thresholding

Thresholding in image segmentation is the most simplest and effective method to separate objects from the background based on image gray level. Thresholding is still used in many applications because of its simplicity and speed. Sezgin et al. [1] provide a survey of the most popular thresholding methods. They categorize thresholding methods based on the information used to six classes: histogram shape-based methods, clustering-based methods, entropy-based methods, object attribute-based methods, the spatial methods and local methods. The Otsu method is considered one of the most popular thresholding techniques. This method divides the image histogram into two classes foreground and background using a threshold that minimizes the variance between these two classes [2]. Bazi et al. [3] use the expectation-maximization algorithm to find the optimal global threshold of the image based on the estimation of the statistical parameters of object and background classes that follow generalized Gaussian distribution. Kamal et al. [4] use a genetic algorithm to generate a threshold from the histogram that has been compressed using wavelet transform. Nakib et al. [5] use the digital fractional differentiation of the image histogram to calculate the threshold of the image. Nakib et al. [6] find the optimal thresholds of the within-class criterion, the entropy and the overall probability of error criterion by using multi-objective optimization approach.

Most of the thresholding techniques depend mainly on the histogram of the image. The shape of the histogram is not the same for all images and most of the methods that depend on the histogram work well when the histogram is bi- or multi-modal. If there is a histogram with not-clear modes, then it will be difficult to detect accurate thresholds as all of these methods are static method, namely without ant ability to adjust to the features of the image. There is a need for a thresholding method that can dynamically adapt to different images. In our method we calculate the threshold of the image based on its feature.

3   Neural Network Segmentation

Because of the large number of different neural architectures used for image segmentation, it is difficult to provide an in-depth review of all proposed schemes. Hence, we will summarize only those works that may be related to our approach. Neural networks have been used in medical image analysis in different ways. Unsupervised clustering neural networks were used to segment CT and MRI brain slices [7]. A three dimensional architecture of contextual constraint-based

Hopfield network [8] was used to segment medical images. An incremental neural network [9] for the segmentation of tissues in ultrasound images by using discrete Fourier transform and discrete cosine transform as a feature vectors. A survey of the image segmentation by relaxation using constraint satisfaction neural network is provided in [10]. Texture features are used with some neural networks to segment different medical images [11]. The features are extracted from a rectangle around the region of interest. The problem with this method is that prior information about the region of interest is required, otherwise manual user interaction is needed.

In our approach, the process is automated as the rectangle around the region of interest that will be used for extracting features is created by the algorithm. Our solution is a simple and efficient neural approach since fast and trainable threshold selection is still necessary for many practical applications.

4   Scale Invariant Feature Transform (SIFT)

SIFT is an approach developed by Lowe [12] as a method for object recognition. It depends on detecting a set of key points of an object and calculating a set of descriptors (features) for these points. These features, invariant to image scale and rotation, can be used to recognize one object from among different objects. The process of detecting key points and calculating descriptors consist of four stages:

Scale-space extrema detection: In this stage the points of interest (key points) are detected. First, Gaussian filters at different scales are generated and then convolved with the image at every scale. The difference of the Gaussian-blurred image are taken and the candidate key points are assigned as maxima/minima of the difference of Gaussians (DoG) $D(x, y, \sigma)$ at multiple scales. To make the scale space of an image, the convolution of a Gaussian, $G(x, y, \sigma)$, with an input image, $I(x, y)$ is taken and defined as a function, $L(x, y, \sigma)$, where

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{1}$$

and, (DoG) is given by:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y). \tag{2}$$

The maxima/minima of the difference of Gaussians (DoG) are calculated by comparing each point with its eight neighbours in the current image and nine neighbours in the scale above and below. The point selected to be key point if it is larger than all of these neighbours or smaller than all of them.

Key point localization: For each candidate point, a measure of stability is calculated to discard points with low contrast or poor location along the edges.

Orientation assignment: local image gradient directions are used to assign one or more orientations to each key point location, to make sure that the candidate points are invariant to orientation.

Key point descriptor: after selecting the candidate points, the local image gradients are measured in the region around each key point. Orientation histograms are created from 4×4 pixel neighbourhoods with 8 bins each, and the magnitude and orientation values of samples in a 16×16 region around the key points are calculated. Where there are $4 \times 4 = 16$ histograms each with 8 bins, the feature vector of each key point has 128 elements.

SIFT has been used in many different applications such as object recognition, image stretching, 3D modelling, gesture recognition, video tracking, match moving, face authentication [13] and to present a self-calibration strategy to estimate intrinsic and extrinsic camera parameters [14]. Many attempts have been made to improve SIFT. Lindeberg [15] proposes a systematic methodology for dealing with the problem of how to select appropriate local scales for further analysis. Tang et al. [16] propose a modification to SIFT algorithm to produce better invariant feature points for image matching under noise. Lopez et al. [17] present a new feature matching algorithm that integrates SIFT local descriptors in the Iterative Closest Point (ICP) scheme to find the appropriate match between repetitive patterns that appear in man-made scenes.

We use SIFT in two different ways. First, we use the SIFT descriptors to train the neural network. Second, we use SIFT to locate the key points inside the image that seems to be the most descriptive points. A set of texture features is extracted from a rectangle around each point.

5   Level Set Segmentation (LS)

The level set method is a tool for modelling time-varying objects as it uses numerical techniques for tracking shape changes. The level set method uses a fixed Cartesian grid to perform numerical computations to follow shapes that change topology involving curves and surfaces without parameterizing these objects [19].

In traditional level set methods [18] a common numerical scheme is used to initialize the function ϕ as a signed distance function before the evolution to overcome the problems of sharp and/or flat shape generated during the curve evolution. This scheme periodically reshapes (or "re-initializes") the function ϕ to be a signed distance function during the evolution. Therefore, the re-initialization process is crucial and cannot be avoided in traditional level set methods [19].

As an attempt to solve the problem of re-initialization, Li et al. [20] suggest to make the level set function close to a signed distance by using internal and external energy terms. Yan et al. [21] used the object's boundary to detect the precise location of the target object and the region information to prevent the boundary leakage problem. Level set methods have been used to segment breast ultrasound images [22] by developing an energy function that employed the differences between the actual and estimated probability densities of the intensities in different regions. The initial contour and the number of iterations need to be assigned for every image which is a time-consuming task.

We used level sets for comparative studies in this paper. We have to initialize the first point that the shape uses to start to evolve. The number of iterations

needs to be assigned for every image to generate perfect results which is computationally expensive.

6   Proposed Approaches

Generally, the proposed idea uses the image characteristics to train a neural network to segment images directly by assigning a threshold $T \in \{0, 1, \ldots, 255\}$. The network is trained with a set of features extracted from training images along with their best-possible threshold as a target of the network. We propose two different approaches to train the neural network. In the first approach, we use the SIFT descriptors calculated for the SIFT key points either for the whole image or within a rectangle around the region of interest (ROI). In the second approach, we use the SIFT method to detect a set of key points within the images either for the whole image or within a rectangle around the ROI. Texture features are extracted from a rectangle around each point.

6.1   Detection Algorithm

All images are processed by a detection algorithm which finds the position of the first seed point (x_s, y_s) inside the ROI. This point is calculated by tracing an $n \times n$ mask (e.g. 10×10) over the image to calculate the sum and the standard deviation of intensities inside every mask, as well as the correlation between the mask and its neighbour. Based on empirical knowledge about the test images, the ROI is rather hypoechoic (low gray-level intensities) and generally exhibits low standard deviations (many cysts and tumours in ultrasound scans of breast are usually hypo- or anechoic). The mask that has minimum sum and minimum standard deviation is considered the mask contains the seed point. The correlation coefficients between the mask and its preceding mask are used to select the mask containing the seed point if the minimum sum and minimum standard deviation are in two different masks. Hence, for every $n \times n$ part of the image the minimum sum, standard deviation and correlation are considered to mark the position of a seed point (x_s, y_s) inside the ROI. This point is used as a center for a rectangle around it.

Generating Rectangle Around ROI The algorithm proceeds by constructing a rectangle around the point (x_s, y_s) as the center of the rectangle. The algorithm starts by finding a region around (x_s, y_s) as a window and enlarges it in small step. This process stops when the standard deviation of one region becomes greater than or less than the standard deviation of the previous region by a certain limit (e.g. 3%) and the last window is considered the rectangle region R around the ROI. These images are stored to be used for extracting feature in the training and testing phases of the proposed methods that use rectangle around the ROI. In most images, the detection algorithm detects a correct point inside the ROI. For some images, it failed to detect a correct point but we nonetheless proceeded with the rectangle generated by the algorithm to verify the overall performance.

6.2 SIFT Descriptors Neural Network (DNN)

Feature Extraction The DNN technique uses the features generated by SIFT for training. The process of extracting features is performed using two different approaches: 1) Extracting features from the whole image (DNN_I). The images I is processed by SIFT to return n points and $128 \times n$ feature matrix, 2) Extracting features from a rectangle around the ROI (DNN_R). R is provided to the SIFT which then returns n points and a $128 \times n$ feature matrix for R.

6.3 SIFT Texture Neural Network (TNN)

Feature Extraction In this approach we extract a set of texture features from a set of key points n assigned by SIFT method either from the whole (TNN_I) image or from a rectangle around the ROI (TNN_R). A 40×40 rectangle is generated around each point and features are calculated from the gray level co-occurrence matrix (GLCM) in the directions 0, 45, 90, and 135 degrees of this rectangle. We use a set of texture features: 1) Contrast: a measure of the intensity difference between a pixel and its neighbours, 2) Correlation: a measure of how correlated a pixel is to its neighbours over the whole image, 3) Energy: the sum of squared elements in the GLCM, 4) Homogeneity: a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal.

We have 4 features in four direction $4 \times 4 = 16$, for $n = 20$, the feature matrix is of size 16×20 where each column represents the features for a point. We use *Matlab* functions "*graycomatrix*" to create the co-occurrence matrix and "*graycoprops*" to calculate the features from the GLCM.

6.4 Training the Neural Network

For the two approaches, a feed-forward backpropagation network was used to learn the set of training images. The network consists of one input layer corresponding to the size of the feature matrix, one hidden layer with 60 nodes and the output layer with one output (=the estimated threshold).

For every training set, five different images are used to train the network. The best-possible thresholds for these five images are assigned as the target of the network. The network is trained using Matlab function "*trainscg*" with desired error set to $\epsilon = 10^{-7}$ to be achieved within a maximum of $N_E = 10^5$ epochs.

6.5 Testing the Neural Network

For DNN and TNN methods, we make two different tests for every new image I. For every approach DNN_I, DNN_R, TNN_I, TNN_R, the global threshold of every image is the average of the thresholds assigned by the network to every point generated for this image by SIFT. The extracted features form every point are the inputs for the neural network which assigns a suitable threshold to each point. Let N_1, N_2, N_3 and N_4 are the points generated by SIFT for DNN_I, DNN_R, TNN_I, TNN_R, respectively. The global threshold for every image is calculated as follows:
$T^* = \frac{1}{|N_j|} \sum_{i=1}^{|N_j|} T_i$ where $j = 1, 2, 3, 4$.

7   Experiments and Results

In this section, a set of 20 breast ultrasound images are employed to train and test the proposed techniques. These 20 images are quite difficult to segment as their maximum achievable segmentation accuracy is 84% (maximum achievable accuracy via pure thresholding if thresholds are optimal). For every approach, DNN and TNN, five of the images are used as a training set and the remaining 15 images are used for testing. This process is repeated 4 times to generate different training sets and investigate the generalization ability of the network. For every training set the process repeated six times and the average of the results was taken. For LS, the process repeated six times as well with number of iterations = 100, 200 and 300 with different sizes of initial contour and the average of the results was taken. For LS method, the algorithm is applied on the 20 images but we divide them into trained images (images that have been used to train the network) and tested images (images used for testing the proposed system) for a better comparison. All results are calculated and recorded in the Tables 1, 2, 3 and 4. The purpose of this experiment is to compare the results from the proposed techniques DNN_I, DNN_R, TNN_I, TNN_R with the results by LS via accuracy calculation using the gold standard images. The following accuracy measures have been employed to verify the performance of the techniques under investigation (A is the binary segmented image and B is the gold standard image): 1) The Dice coefficient D of two binary images A and B: $D(A, B) = \frac{2*|A \cap B|}{|A|+|B|}$, 2) The standard deviation σ_D of the Dice coefficient, 3) The 95% confidence interval (CI) of D, 4) The average of segmentation accuracy J is calculated using the area overlap (also called Jaccard Index): $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$, 5) The standard deviation σ of the average of accuracy, and 6) The 95% confidence interval (CI) of J.

It is obvious from tables 1, 2,3 and 4 that the proposed algorithms, DNN_I, DNN_R, TNN_I and TNN_R perform more accurately compared to LS method. Generally, it is apparent from the tables that the proposed algorithms have the highest accuracies compared with the LS algorithm for all training sets. The proposed approaches have the highest average segmentation accuracy and the lowest standard deviation for all cases. Table 5 shows a summary of the results of the four training sets with focusing on the average of segmentation accuracy J and the 95% confidence interval CI. It can be seen that the proposed approach has the highest average segmentation accuracy and the shortest confidence intervals as well, which means that the proposed approach can provide more accurate and more consistent results. For example, in the third training set, the average accuracy of the proposed system raised from 66% (LS) to 74% (TNN_I), 76% (TNN_R), 80% (DNN_I) and 81% (DNN_R). Moreover, the confidence interval is pushed higher from 55%–78% (LS) to 66%–84%(TNN_I), 67%–85% (TNN_R), 74%–86% (DNN_I)and 74%–87% (DNN_R).

Table 1. The first training set: The Jaccard Index J, Dice value D and their standard deviations σ_J, σ_D, the 95%-confidence interval CI of J and of D.

Metrics	LS		Train	TNN_I	TNN_R	DNN_I	DNN_R
	Train	Test					
D	72	79	90	84	84	87	89
σ_D	± 19	± 18	±5	±15	±15	±11	±7
CI_D	46-96	68-89	84-97	76-93	75-92	80-94	85-93
J	59	69	82	75	74	78	80
σ_J	± 25	± 22	± 9	± 18	± 19	± 15	± 10
CI_J	28-91	56-81	72-94	65-86	64-85	69-87	74-86

Table 2. The second training set: The Jaccard Index J, Dice value D and their standard deviations σ_J, σ_D, the 95%-confidence interval CI of J and of D.

Metrics	LS		Train	TNN_I	TNN_R	DNN_I	DNN_R
	Train	Test					
D	75	77	90	79	87	87	88
σ_D	± 19	± 18	± 6	± 22	± 13	± 13	± 9
CI_D	50-97	68-89	82-98	66-91	79-94	79-94	82-94
J	63	67	82	69	78	78	79
σ_J	± 23	± 22	± 10	± 25	± 17	± 16	± 13
CI_J	33-92	54-80	69-95	55-83	68-88	68-87	71-87

8   Conclusions

Thresholding could be an efficient technique for image segmentation because of its simplicity and fastness. However, most thresholding techniques treat all images with the same static nature regardless of the properties of the image. These methods may generate good results for some images but they are not a universal solution. Intelligent segmentation by training a neural network to generate the threshold for the images based on their features seems to be a more flexible solution. In this paper, we used a neural network to segment images based on their characteristics. The network was trained using a set of features

Table 3. The third training set: The Jaccard Index J, Dice value D and their standard deviations σ_J, σ_D, the 95%-confidence interval CI of J and of D.

Metrics	LS		Train	TNN_I	TNN_R	DNN_I	DNN_R
	Train	Test					
D	76	78	90	84	86	89	89
σ_D	± 25	± 16	± 7	± 12	± 11	± 6	±6
CI_D	44-100	68-87	80-99	78-91	79-93	85-93	85-93
J	66	66	82	74	76	80	81
σ_J	± 30	± 20	±12	± 16	± 15	± 10	± 10
CI_J	28-97	55-78	67-97	66-84	67-85	74-86	74-87

Table 4. The fourth training set: The Jaccard Index J, Dice value D and their standard deviations σ_J,σ_D, the 95%-confidence interval CI of J and of D.

Metrics	LS		Train	TNN_I	TNN_R	DNN_I	DNN_R
	Train	Test					
D	69	80	92	78	86	87	88
σ_D	± 23	± 16	±3	±20	± 13	± 11	± 8
CI_D	40-97	70-90	87-97	67-90	78-93	81-94	84-93
J	57	69	84	68	77	79	80
σ_J	± 27	± 20	± 6	± 23	± 17	± 14	± 12
CI_J	22-92	58-81	76-93	55-82	68-87	71-88	73-86

Table 5. Summary of the results for the accuracy J and its 95%-confidence interval.

	1st set		2nd set		3rd Set		4th Set	
Method	J	CI	J	CI	J	CI	J	CI
LS	69%	56% -81%	67%	55% -80%	66%	55% -78%	69%	58% -81%
TNN_I	75%	65% -86%	69%	55% -83%	74%	66% -84%	68%	55% -82%
TNN_R	74%	64%-85%	78%	68%-88%	76%	67%-85%	77%	68%-87%
DNN_I	78%	69%-87%	78%	68%-87%	80%	74%-86%	79%	71%-88%
DNN_R	80%	74%-86%	79%	71%-87%	81%	74%-87%	80%	73%-86%

extracted from the image in different ways. The SIFT method was used to detect key points and provide descriptors. These approaches seems to be simple, efficient and accurate for designing dynamic segmentation techniques for a difficult real-world problem such as breast ultrasound imaging.

References

1. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. Electronic Imaging 13, 146–165 (2004)
2. Otsu, N.: A threshold selection method from gray-level histograms. Automatica 11, 285–296 (1975)
3. Bazi, Y., Bruzzone, L., Melgani, F.:Image thresholding based on the EM algorithm and the generalized Gaussian distribution. Pattern Recognition 40, 619 – 634 (2007)
4. Hammouche, K., Diaf, M., Siarry, P.: A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. Computer Vision and Image Understanding 109, 163–175 (2008)
5. Nakib, A., Oulhadj, H., Siarry, P.:Fractional differentiation and non-Pareto multiobjective optimization for image thresholding. Engineering Applications of Artificial Intelligence 22, 236–249 (2009)
6. Nakib, A., Oulhadj, H., Siarry, P.:Image histogram thresholding based on multi-objective optimization. Signal Processing 87, 2516–2534 (2007)
7. Ahmed, M.N., Farag, A.A.: Two-stage neural network for volume segmentation of medical images. Pattern Recognition Letters 18, 1143–1151 (1997)

8. Chang, C.Y., Chung, P.C.: medical image segmentation using a contextualconstraint- based Hopfield neural cube. Image and Vision Computing 19, 669–678, (2001)
9. Kurnaz, M.N., Dokur, Z., Ölmez, T.: An incremental neural network for tissue segmentation in ultrasound images. Computer Methods and Programs in Biomedicine 85, 187–195 (2007)
10. Kurugollu, F., Sankur, B., Harmanci, A.: Image segmentation by relaxation using constraint satisfaction neural network. Image & Vision Comp. 20, 483–497 (2002).
11. Sharma, N., Ray, A.K., Sharma, S., Shukla, KK, Pradhan, S., Aggarwal, L.M.:Segmentation and classification of medical images using texture-primitive features: Application of BAM-type artificial neural network. Medical physics/Association of Medical Physicists of India 33, 119 (2008).
12. Lowe, D.G.: Object recognition from local scale-invariant features. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)
13. Bicego, M., Lagorio, A., Grosso, E., Tistarelli, M.: On the use of SIFT features for face authentication. In: Computer Vision and Pattern Recognition Workshops, pp.35 (2006)
14. Yun, J.H., Park, R.H.: Self-calibration with two views using the scale-invariant feature transform. Advances in Visual Computing, 589–598 (2006)
15. Lindeberg, T.: Feature Detection with Automatic Scale Selection. Computer Vision 30, 79– 116 (1998)
16. Tang, C.Y., Wu, Y.L., Hor, M.K., Wang, W.H.: Modified SIFT descriptor for image matching under interference. In: ICMLC, Vol. 6, pp. 3294–3300 (2008)
17. Lemuz-López, R., Arias-Estrada, M.: Iterative closest SIFT formulation for robust feature matching. Advances in Visual Computing, 502–513 (2006)
18. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. IJCV 22, 61–79 (1997).
19. Osher, S., Sethian, J.A.: Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. Computational Physics 79, 12–49 (1998)
20. Li, C., Xu, C., Gui, C., Fox, M.D.: Level Set Evolution Without Re-initialization: A New Variational Formulation. Computer Vision and Pattern Recognition (2005)
21. Zhang, Y., Matuszewski, B.J., Shark, L.K., Moore, C.J.: Medical image segmentation using new hybrid level-set method. In: Medivis 2008, pp. 71–76 (2008)
22. Liu, B., Cheng, HD., Huang, J., Tian, J., Tang, X., Liu, J.: Probability density difference-based active contour for ultrasound image segmentation. Pattern Recognition 43, 2028–2042 (2010)