COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 113 (2014) 202–209

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

Lung cancer classification using neural networks for CT images

*Jinsa Kuruvilla**, *K. Gunavathi*

ECE Department, PSG College of Technology, Coimbatore 641004, India

ARTICLE INFO

Article history:
Received 14 March 2013
Received in revised form
12 September 2013
Accepted 8 October 2013

Keywords:
Computed tomography
Skewness
Kurtosis
Neural network

ABSTRACT

Early detection of cancer is the most promising way to enhance a patient's chance for survival. This paper presents a computer aided classification method in computed tomography (CT) images of lungs developed using artificial neural network. The entire lung is segmented from the CT images and the parameters are calculated from the segmented image. The statistical parameters like mean, standard deviation, skewness, kurtosis, fifth central moment and sixth central moment are used for classification. The classification process is done by feed forward and feed forward back propagation neural networks. Compared to feed forward networks the feed forward back propagation network gives better classification. The parameter skewness gives the maximum classification accuracy. Among the already available thirteen training functions of back propagation neural network, the Traingdx function gives the maximum classification accuracy of 91.1%. Two new training functions are proposed in this paper. The results show that the proposed training function 1 gives an accuracy of 93.3%, specificity of 100% and sensitivity of 91.4% and a mean square error of 0.998. The proposed training function 2 gives a classification accuracy of 93.3% and minimum mean square error of 0.0942.

1. Introduction

Lung cancer is the leading cause of cancer deaths in both women and men. It is estimated that 1.2 million people are diagnosed with this disease every year (12.3% of the total number of cancer diagnosed), and about 1.1 million people are dying of this disease yearly (17.8% of the total cancer death) [1]. The survival rate is higher if the cancer is detected at early stages. The early detection of lung cancer is not an easy task. About 80% patients are diagnosed correctly at the middle or advanced stage of cancer [2]. Computer-aided diagnosis system is very helpful for radiologist in detection and diagnosing abnormalities earlier and faster [3]. The computer aided diagnosis is a second opinion for radiologists before suggesting a biopsy test [4]. In recent research literature, it is observed that principles of neural networks have been widely used for the detection of lung cancer in medical images [5].

For classification of lung cancer, few methods based on neural network have been reported in the literature. Abdulla et al. [6] proposed a computer aided diagnosis based on artificial neural networks for classification of lung cancer. The features used for classification are area, perimeter and shape. The maximum classification accuracy obtained is 90%. Camarlinghi et al. [7] proposed a computer-aided detection algorithm for automatic lung nodule identification. The sensitivity obtained is 80% with 3 FP/scan. Al-Kadi et al. [8] proposed a classification method based on fractal texture features. The classification accuracy obtained is 83.3%. van Ginneken et al. [9] compared and combined six computer aided detection algorithms for pulmonary nodules. The combination of six algorithms is able to detect 80% of all nodules at the expense

* *Corresponding author*. Tel.: +91 9946104098.
E-mail addresses: jinsa_k@yahoo.com (J. Kuruvilla), kgunavathi2000@yahoo.com (K. Gunavathi).
0169-2607/$ – see front matter
http://dx.doi.org/10.1016/j.cmpb.2013.10.011

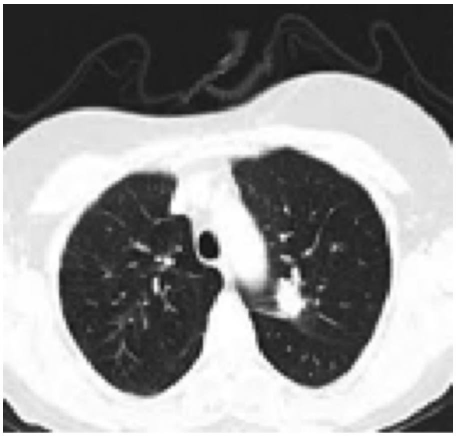

Fig. 1 – CT image of lungs with cancer.

Fig. 2 – Binary image.

Fig. 3 – Morphological opening output.

of only two false positive detections per scan and 65% of all nodules with only 0.5 false positives. Cascio et al. [10] proposed a computer-aided detection (CAD) system for the selection of lung nodules in computer tomography (CT) images. The detection rate of the system is 88.5% with 6.6 FPs/CT on 15 CT scans. A reduction to 2.47 FPs/CT is achieved at 80% efficiency.

2. Segmentation

The images are collected from a database of Lung Image Database Consortium (LIDC) and also from reputed hospital. CT images of 155 patients are collected including both men and women. The average age of the patients considered is 64.2 years (age of the youngest patient is 18 years and the oldest patient is 85 years). The low dose CT scan images are obtained at kilo voltage peak distribution of 120–140 KVp with a current varying from 25 to 40 mAs depending upon the age of the patient. The reconstruction diameter varies from 260 to 400 mm with a slice thickness of 0.75–1.25 mm. The total number of 110 nodules of size >3 mm are considered in this study and the nodules are referred by two radiologists. The final response was the consensus decision of the two radiologists. The two radiologists have referred the LIDC CT scans without considering the annotations available in the LIDC database. Both primary and secondary stage cancer nodules (classified by two radiologists depending on the size of the nodules) with four different kinds of nodules like Well-circumscribed nodules, Vascularized nodules, Juxta-pleural nodules and Pleural-tail nodules are considered in the work [11]. Fig. 1 shows the CT image of lungs with cancerous region.

The lung is segmented from the CT images using morphological operations. The gray scale image is first converted to binary image. All the pixels in the input image with a intensity greater than a threshold level is replaced with value '1' and all pixel values with a intensity less than threshold level is replaced with value '0'. The threshold level is calculated by Otsu method [12]. The Otsu method chooses the threshold level to minimize the intraclass variance of the black and white pixels. Fig. 2 gives the grayscale to binary converted image.

The morphological opening operation is performed to the binary image with a structuring element. The structuring is a shape, used to probe or interact with a given image, with the purpose of drawing conclusions on how this shape fits or misses the shapes in the image. The structuring element used is 'periodicline'. It is a flat structuring element containing $2 \times (P + 1)$ members. The value of 'P' specifies the size of the structuring element. The P value is selected as 2. One structuring element is located at the origin. Fig. 3 shows the output after morphological operation. The image is then inverted and clear border operation is performed. The clear border operation suppresses structures that are lighter than their surroundings and that are connected to the border of the image. The segmentation method uses only morphological operations and an average of 98% of images is segmented correctly. The segmented images are independently referred by two radiologists. Main advantage of morphological operation is their speed and simplicity of implication. Fig. 4 shows the final segmented output.

3. Statistical parameters

The statistical parameters are extracted from the region of interest. The region of interest is the segmented single slices containing 2 lungs. The parameters considered are mean, standard deviation, skewness and kurtosis [13]. In this paper

Fig. 4 – Segmented output.

the higher order moments like fifth central moment and sixth central moment are also considered.

3.1. *Mean*

The mean, μ of the pixel values in the defined window, estimates the value in the image in which central clustering occurs.

$$\mu = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} p(i, j)$$

where $p(i, j)$ is the intensity value of the pixel at the point (i, j). $M \times N$ is the size of the image.

3.2. *Standard deviation*

The standard deviation, σ is the estimate of the mean square deviation of gray pixel value $p(i, j)$ from its mean value μ.

$$\sigma = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(p(i, j) - \mu)^2}$$

3.3. *Skewness*

Skewness, S characterizes the degree of asymmetry of a pixel distribution in the specified window around its mean. Skewness is a pure number that characterizes only the shape of the distribution.

$$S = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[\frac{p(i, j) - \mu}{\sigma}\right]^3$$

3.4. *Kurtosis*

Kurtosis, K measures the peakness or flatness of a distribution relative to a normal distribution.

$$K = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[\frac{p(i, j) - \mu}{\sigma}\right]^4$$

3.5. *Fifth and sixth central moment*

The fifth and sixth central moments are given respectively by

$$\text{Fifth central moment} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[\frac{p(i, j) - \mu}{\sigma}\right]^5$$

$$\text{Sixth central moment} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[\frac{p(i, j) - \mu}{\sigma}\right]^6$$

4. Artificial neural network

The neural nets can be trained to perform pattern classification [14]. The simplest neural network used for pattern classification consists of a layer of input unit and a single output unit. Both feed forward and feed forward back propagation neural networks are used for classification.

4.1. *Feed forward neural network*

In feed forward neural networks information always moves in one direction only, there is no feedback. The information moves forward from input layer through hidden layer to the output layer. The networks used are Hebb, Perceptron, Adaline and Madaline networks. In Hebb network learning is done by modification of the weights of the neurons. The weight is information used by neural network to solve a problem. If two interconnected neurons are both 'on' at the same time, then the weight between those neurons are increased [15]. The Perceptron network is supervised classifier for classifying an input into one of two possible outputs. It is a type of linear classifier. The classification algorithm makes its predictions based on a linear predictor function combining a set of weights with the feature vector describing a given input. Both bias and threshold are needed in this network. The Adaline (Adaptive Linear Neuron) network uses bipolar activations for its input signals and target output. The weights on the connection from the input units to the adaline network are adjustable. The network has a bias, which acts like an adjustable weight on a connection from a unit whose activation is always 1. A Madaline network consists of adalines arranged in a multilayer net. It is a two layer neural network with a set of adalines in parallel as its input layer and a single processing element in its output layer.

4.2. *Feed forward back propagation neural network*

The back propagation is a systematic method of training multilayer neural networks in a supervised manner. The back propagation method, also known as the error back propagation algorithm, is based on the error-correction learning rule. The back propagation network consists of atleast three layers of units: an input layer, at least one intermediate hidden layer and one output layer. The units are connected in feed forward fashion with inputs units connected to the hidden layer units and the hidden layer units are connected to the output layer units. An input pattern is forwarded to the output through input to hidden and hidden to output weights. The output of the network is the classification decision.

4.3. *Types of training functions used for classification*

The back propagation neural networks are trained with thirteen training algorithms or functions [16]. The training functions used are Gradient descent back propagation (traingd), Gradient descent with variable learning rate (traingda), Gradient descent with momentum (traingdm), Gradient descent with variable learning rate and momentum (traingdx), Resilient back propagation (trainrp), Conjugate Gradient Algorithms (traincgf, traincgp, traincgb, trainscg), Quasi Newton BFGS (trainbfg), One Step Secant Algorithm (trainoss), Levenberg–Marquardt (trainlm) and Automated Regularization (trainbr).

4.4. *Gradient descent back propagation (traingd)*

Traingd is a training function which updates weight and bias values according to gradient descent. A network can be trained when its weight, net input, and transfer functions have derivative functions. The weights and biases are updated in the direction of the negative gradient of the performance function [17]. The learning rate '*lr*' is multiplied with the negative of the gradient to determine the changes to the weights and biases. Back propagation is used to calculate derivatives of performance (*dperf*) with respect to the weight and bias variables X. Each variable is adjusted according to gradient descent:

$$dX = lr \times \frac{dprf}{dX}$$

4.5. *Gradient descent with variable learning rate (traingda)*

Traingda is a training function that updates weight and bias values depending on the gradient descent with adaptive learning rate. The performance of the back propagation neural network is sensitive to the learning rate. It is not practical to determine the optimal setting for the learning rate before training. The optimal learning rate changes during the training process, as the algorithm moves across the performance surface. Any network can be trained when its weight, net input, and transfer functions are derivative functions. Back propagation algorithm is used to calculate derivatives of performance (dperf) with respect to the weight and bias variables X. If performance decreases toward the goal, then the learning rate is increased by a constant factor. If performance increases by more than the maximum performance fixed, the learning rate is decreased by a constant factor.

4.6. *Gradient descent with momentum (traingdm)*

Traingdm is a network training function that updates weight and bias values according to gradient descent with momentum. Each variable is adjusted according to gradient descent with momentum,

$$dX = mc \times dXprev + lr \times (1 - mc) \times \frac{dperf}{dX}$$

where '*dXprev*' is the previous change to the weight or bias, '*mc*' is momentum constant and '*lr*' is learning rate.

4.7. *Gradient descent with variable learning rate and momentum (traingdx)*

Traingdx is a network training function that updates weight and bias values according to gradient descent momentum and adaptive learning rate. Each variable is adjusted according to the gradient descent with momentum [18]. The performance can be increased if the learning rate is allowed to change during the training process. The initial network output and error are calculated first. At each epoch new weights and biases are calculated using the current learning rate. New outputs and errors are then calculated. Each variable is adjusted according to the gradient descent with momentum.

$$dX = mc \times dXprev + lr \times mc \times \frac{dperf}{dX}$$

When momentum is considered, if the new error exceeds the old error by more than a predefined ratio, the new weights and biases are discarded.

4.8. *Resilient back propagation (trainrp)*

Trainrp is a network training function that updates weight and bias values according to the resilient backpropagation algorithm [19]. The purpose of the resilient backpropagation (Rprop) training algorithm is to eliminate the effects of the magnitudes of the partial derivatives. The sign of the derivative determines the direction of the weight update. The magnitude of the derivative has no effect on the weight update. If the derivative is zero, then the update value remains the same. Whenever the weights are oscillating, the weight change is reduced. If the weight continues to change in the same direction for several iterations, then the magnitude of the weight change is increased.

4.9. *Conjugate gradient algorithms*

In the conjugate gradient algorithms a search is performed along conjugate directions. In conjugate gradient algorithms, the step size is adjusted at each iteration [20]. A search is made along the conjugate gradient direction to determine the step size that minimizes the performance function along that line.

(a) *Fletcher-Reeves Update (traincgf)*: The steepest descent direction (negative of the gradient) is searched by the first iteration.

$$P_0 = -g_0$$

where P_0 is the initial search gradient and g_0 is the initial gradient. The optimal distance to move along the current search direction is obtained by line search.

$$X_{k+1} = X_k + \alpha_k p_k$$

where X_{k+1} is the next weight vector, X_k is the current weight vector, α_k is the learning rate and p_k is the current search direction. The next search direction is found so that it is conjugate to the previous search directions. The new search direction is determined by combining the new steepest descent direction with the previous search direction.

$$p_k = -g_k + \beta_k p_{k-1}$$

where p_{k-1} is the previous search direction and β_k is a constant.

For Fletcher Reeves update, the constant β_k is obtained by

$$\beta_k = \frac{g_k^T g_k}{g_{k-t}^T g_{k-1}}$$

This is the ratio of the square of the current gradient to the square of the previous gradient. Each variable is adjusted according to the following:

$$X = X + a \times dX$$

where dX is the search direction. The parameter 'a' is selected to minimize the performance along the search direction.

(b) Polak and Ribiere (traincgp)

In Polak and Ribiere update, the constant β_k is obtained by

$$\beta_k = \frac{\Delta g_k^T g_k}{g_{k-t}^T g_{k-1}}$$

This is the inner product of the previous change in the gradient with the current gradient divided by the square of the previous gradient.

(c) Powell and Beale Restarts (traincgb)

The search direction is periodically reset to the negative of the gradient for all conjugate gradient algorithms [21]. The standard reset point occurs when the number of iterations is equal to the number of network parameters. The Powell and Beale proposed a new reset method to increase the efficiency of the training process. This technique restarts if there is very little orthogonality left between the current gradient and the previous gradient. This is tested with the following inequality:

$$|g_{k-1}^T g_k| \geq 0.2 \times ||g_k||^2$$

If the condition is satisfied, the search direction is reset to the negative of the gradient.

(d) Scaled Conjugate Gradient Algorithm (trainscg)

Each of the conjugate gradient algorithms requires a line search at each iteration. This line search is computationally expensive. It requires that the network response to all training inputs be computed several times for each search. The scaled conjugate gradient algorithm (SCG), was designed to avoid the time-consuming line search.

4.10. *Quasi Newton BFGS (trainbfg)*

Trainbfg is a network training function that updates weight and bias values according to the BFGS quasi-Newton method. The Newton's method has faster optimization than conjugate gradient method. The convergence of Newton's method is faster than conjugate gradient methods. The basic weight update of Newton's method is given by

$$W_{k+1} = W_k - A_k^{-1} g_k$$

where A_k is the Hessian matrix of the performance index in the current values of the weights and biases. As A_k value increases the complexity and time consumption in computing W_{k+1} increases. It is complex and expensive to compute the Hessian matrix. Quasi Newton method is based on Newton's method but does not require the calculation of Hessian matrix. An approximation of Hessian matrix is updated at each iteration of the algorithm. The update is computed as a function of the gradient.

4.11. *One step secant algorithm (trainoss)*

Trainoss is a network training function that updates weight and bias values according to the one step secant method [22]. The BFGS algorithm requires more storage and computation in each iteration than the conjugate gradient algorithms. The one step secant (OSS) method is an attempt to bridge the gap between the conjugate gradient algorithms and the quasi-Newton algorithms. This algorithm does not store the complete Hessian matrix. It assumes that at each iteration, the previous Hessian was the identity matrix. The additional advantage is that the new search direction can be calculated without computing a matrix inverse.

4.12. *Levenberg–Marquardt (trainlm)*

Trainlm is a network training function that updates weight and bias states according to Levenberg–Marquardt optimization [23]. Trainlm is the fastest backpropagation algorithm, and is highly recommended as a first choice supervised algorithm, although it does require more memory than other algorithms. When the performance function has the form of a sum of squares, then the Hessian matrix can be approximated as

$$H = J^T J$$

The gradient can be computed as

$$g = J^T e$$

where J is the Jacobian matrix that contains first derivatives of network errors with respect to the weights and biases and 'e' is a vector of network errors. The Jacobian matrix can be computed using back propagation algorithm. It is much less complex than computing the Hessian matrix.

The Levenberg–Marquardt algorithm uses an approximation to the Hessian matrix.

$$X_{k+1} = X_k - (J^T J + \mu I)^{-1} J^T e$$

where I is the identity matrix.

4.13. *Automated regularization (trainbr)*

Trainbr is a network training function that updates the weight and bias values according to Levenberg–Marquardt optimization. It minimizes a combination of squared errors and weights and, then determines the correct combination so as to produce a network which generalizes well. The process is called Bayesian regularization [24]. It is desirable to determine the optimal regularization parameters in an automated fashion. The weights and biases of the network are assumed to be random variables with specified distributions. The regularization parameters are related to the unknown variances associated with these distributions. The Bayesian regularization takes place within the Levenberg–Marquardt algorithm. Backpropagation is used to calculate the Jacobian *jX* of performance PERF with respect to the weight and bias variables *X*. Each variable is adjusted according to Levenberg–Marquardt

$$jj = jX \times jX$$

$$je = jX \times E$$

$$dX = -\frac{jj + I \times mu}{je}$$

where *E* is all errors and *I* is the identity matrix.

Two new training functions are proposed in this paper.

4.14. *Proposed training functions*

4.14.1. *Training function 1*

The momentum value and the learning rate influence the classification accuracy and the mean square error of the neural network. A new training function which includes the momentum factor and the learning rate is proposed. In the training function 1 each variable is adjusted according to the gradient descent with momentum given by

$$dX = mc \times dXprev + lr \times (1 - mc) \times mc \times \frac{dperf}{dX}$$

where '*dXprev*' is the previous change to the weight or bias, '*mc*' is momentum constant, '*lr*' is learning rate and '*dperf*' is the derivative of performance with respect to the weight and bias variables *X*. The classification accuracy is increased by the training function 1.

4.14.2. *Training function 2*

The training function 1 is modified to reduce the mean square error. In the training function 2 each variable is adjusted according to the gradient descent with momentum given by

$$dX = 3.7 \times mc \times dXprev + lr \times (1 - mc) \times mc \times \frac{dperf}{dx}$$

where '*dXprev*' is the previous change to the weight or bias, '*mc*' is momentum constant, '*lr*' is learning rate and '*dperf*' is the derivative of performance with respect to the weight and bias variables *X*. The mean square error is minimum for training function 2.

5. Experimental results

The statistical parameters are calculated for segmented single slices containing 2 lungs and given as input to the neural network. The training set for the neural network consists of 70% of the total images and the testing set is 30% of the total images. Images for training and testing refer to the segmented single slices containing 2 lungs. The sensitivity, specificity and accuracy are calculated for each network.

5.1. *Sensitivity*

It measures the proportion of actual positives which are correctly identified. That is the percentage of segmented slices containing cancerous nodule is correctly classified as cancerous.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

True Positive (TP): Segmented slice containing cancer nodule is classified as cancerous.
False Positive (FP): Segmented slice without cancer nodule is classified as cancerous.
True Negative (TN): Segmented slice without cancer nodule is classified as non-cancerous.
False Negative (FN): Segmented slice containing cancer nodule is classified as non-cancer

5.2. *Specificity*

It measures the proportion of negatives which are correctly identified. The percentage of segmented slices without cancerous nodule is correctly identified as non cancerous.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

5.3. *Accuracy*

Accuracy is a statistical measure of how well a classifier correctly identifies or excludes a condition. The accuracy is the proportion of true results (both true positive and true negative) in the population.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

The result shows that the parameter skewness gives maximum classification accuracy. The higher order moments are considered, but no significant improvement in the classification accuracy is observed. Compared to the feed forward networks, the classification accuracy of back propagation neural network is more. The classification accuracy of feed forward networks and back propagation network is shown in Fig. 5. All the thirteen training functions of the back propagation network are trained with different momentum factor and learning rate. The momentum and learning rate are varied from 0.1 to 0.9. Keeping one of these parameters constant and varying the other, the performance of the network is studied. The best classification accuracy and least mean square error of different training functions for different momentum

Table 1 – Best classification of 13 training algorithms.

Training functions	Momentum	Learning rate	Classification accuracy	Mean square error
Traingd	0.4	0.6	86.7	0.1242
Traingda	0.3	0.5	88.89	0.118
Traingdm	0.4	0.7	88.89	0.115
Traingdx	0.3	0.7	**91.11**[a]	**0.112**[a]
Traincgf	0.3	0.5	75.5	0.164
Traincgb	0.6	0.3	73.3	0.198
Traincgp	0.4	0.8	75.56	0.162
Trainlm	NA	0.8	82.2	0.144
Trainoss	0.6	0.6	73.3	0.196
Trainrp	0.4	0.7	80	0.154
Trainscg	0.6	0.3	73.3	0.197
Trainbfg	NA	0.5	80	0.156
Trainbr	0.4	0.6	88.89	0.115

[a] The maximum classification rate obtained among the 13 training functions.

and learning rate are shown in Table 1. Table 1 shows that the training function Traingdx gives the maximum classification accuracy of 91.11%. The Specificity, Sensitivity and Accuracy of Traingdx and the proposed training function 1 with momentum 0.3 and learning rate 0.7 are shown in Fig. 6. The classification accuracy and mean square error of Traingdx,

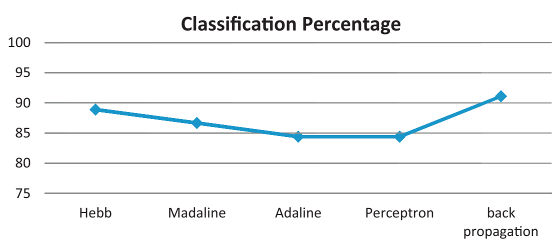

Fig. 5 – Classification accuracy of feed forward networks and back propagation network.

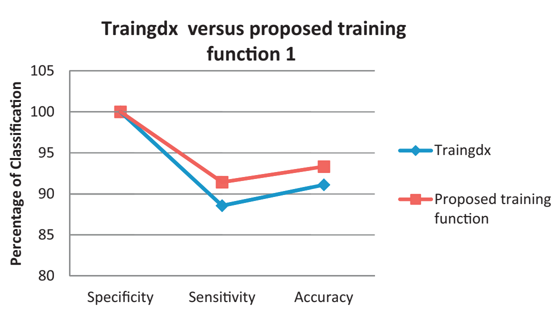

Fig. 6 – Traingdx versus proposed training function 1.

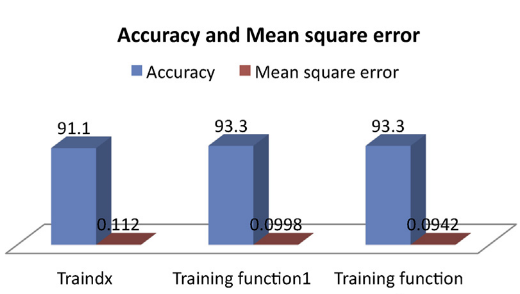

Fig. 7 – Classification accuracy and mean square error of Traindx, training algorithm 1 and training algorithm 2.

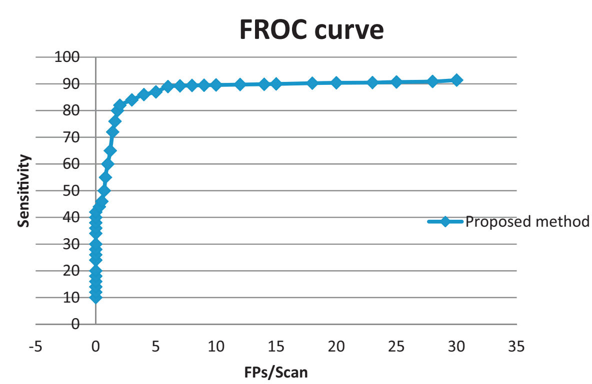

Fig. 8 – FROC curve.

training function 1 and training function 2 with momentum 0.3 and learning rate 0.7 are shown in Fig. 7. The performance of the system is analyzed by the FROC (Free-response Receiver Operating Characteristic) curve shown in Fig. 8. The sensitivity is measured as the percentage of segmented slices containing cancerous nodule is correctly classified as cancerous. The FROC curve shows that the sensitivity is 82% with 2 FPs/scan and the sensitivity increased to 90% with 15 FPs/scan and to 91.4% with 30 FPs/scan.

6. Conclusion

A computer aided segmentation and classification method is proposed. Morphological operations are used for segmentation and classification is done by different neural networks. The region of interest is the segmented single slices containing 2 lungs. The statistical parameters are used as features for classification. Compared to the statistical parameters like mean, standard deviation, kurtosis, fifth central moment and sixth central moment, skewness gives the maximum classification accuracy. There is 5–8% increase in classification accuracy for skewness. Among the already available thirteen training functions, the traingdx training function gives the maximum classification accuracy of 91.11%. The proposed training function 1 gives classification accuracy of 93.3% with a specificity of 100% and sensitivity of 91.4%. The proposed training function 2 gives classification accuracy of 93.3% and mean square error of 0.0942. The performance of the proposed CAD system is good with a sensitivity of 82% with 2 FPs/scan, which is increased to 91.4% with 30 FPs/scan. The sensitivity is measured as the percentage of segmented slices containing cancerous nodule is correctly classified as cancerous. The misclassification occurred in the images where the cancer nodule is located near the pleural side of the lungs.

Conflict of interest

The authors do not have any conflicts of interest, and disclose any financial and personal relationships with other people or organizations that could inappropriately influence (bias) the presented work.

REFERENCES

[1] D.M. Parkin, Global cancer statistics in the year 2000, Lancet Oncology 2 (2001) 533–543.
[2] A. Motohiro, H. Ueda, H. Komatsu, N. Yanai, T. Mori, Prognosis of non-surgically treated, clinical stage I lung cancer patients in Japan, Lung Cancer 36 (2002) 65–69.
[3] R.N. Strickland, Tumor detection in non stationary backgrounds, IEEE Transactions on Medical Imaging 13 (June) (1994) 491–499.
[4] S.B. Lo, S.L. Lou, J.S. Lin, M.T. Freedman, S.K. Mun, Artificial convolution neural network techniques and applications for lung nodule detection, IEEE Transactions on Medical Imaging 14 (August) (1995) 711–718.
[5] G. Coppini, S. Diciotti, M. Falchini, N. Villari, G. Valli, Neural networks for computer aided diagnosis: detection of lung nodules in chest radiograms, IEEE Transactions on Information Technology in Biomedicine 4 (2003) 344–357.
[6] A.A. Abdulla, S.M. Shaharum, Lung cancer cell classification method using artificial neural network, Information Engineering Letters 2 (March) (2012) 50–58.
[7] N. Camarlinghi, et al., Combination of computer-aided detection algorithms for automatic lung nodule identification, International Journal of Computer Assisted Radiology and Surgery 7 (2012) 455–464.
[8] O.S. Al-Kadi, D. Watson, Texture analysis of aggressive and non aggressive lung tumor CE CT images, IEEE Transactions on Biomedical Engineering 55 (2008) 1822–1830.
[9] B. van Ginneken, et al., Comparing and combining algorithms for computer-aided detection of pulmonary nodules in computed tomography scans: the ANODE09 study, Medical Image Analysis 14 (2010) 707–722.
[10] R. Bellotti, D. Cascio, et al., A CAD system for nodule detection in low-dose lung CTs based on region growing and a new active contour model, International Journal of Medical Physics Research and Practice 34 (2007) 4901–4911.
[11] W.J. Kostis, A.P. Reeves, D.F. Yankelevitz, C.I. Henschke, Three dimensional segmentation and growth-rate estimation of small pulmonary nodules in helical CT images, IEEE Transactions on Medical Imaging 22 (2003) 1259–1274.
[12] R.C. Gonzalez, R.E. Wood, Digital Image Processing, 3ed ed., Pearson Prentice Hall, 2008.
[13] S.A. Patil, V.R. Udupi, C.D. Kane, A.I. Wasif, J.V. Desai, A.N. Jadhav, Geometrical and texture feature estimation of lung cancer and TB image using chest X-ray database, in: IEEE, 2009.
[14] M.G. Penedo, M.J. Carreira, A. Mosquera, Computer-aided diagnosis a neural-network-based approach to lung nodule detection, IEEE Transactions On Medical Imaging 17 (1998) 872–880.
[15] S.N. Sivanandan, S. Sumathi, S.N. Deepa, Introduction to Neural Networks Using Matlab, Tata McGraw Hill Publishing Company Limited, 2006.
[16] F. Paulin, A. Santhakumaran, Back propagation neural network by comparing hidden neurons: case study on breast cancer diagnosis, International Journal of Computer Applications 2 (June) (2010) 40–44.
[17] J. Ramesh, K. Gunavathi, et al., Fault classification in phase-locked loops using back propagation neural networks, ETRI, Journal 30 (August (4)) (2008) 546–554.
[18] M.T. Hagan, H.B. Demuth, M.H. Beale, Neural Network Design, PWS Publishing, Boston, MA, 1996.
[19] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the RPROP algorithm, in: Proceedings of the IEEE International Conference on Neural Networks, 1993.
[20] C. Charalambous, Conjugate gradient algorithm for efficient training of artificial neural networks, IEEE Proceedings 139 (3) (1992) 301–310.
[21] M.F. Moller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (1993) 525–533.
[22] R. Battiti, First and second order methods for learning: Between steepest descent and Newton's method, Neural Computation 4 (2) (1992) 141–166.
[23] M.T. Hagan, M. Menhaj, Training feed-forward networks with the Marquardt algorithm, IEEE Transactions on Neural Networks 5 (6) (1994) 989–993.
[24] O. De Jesús, M.T. Hagan, Back propagation algorithms for a broad class of dynamic networks, IEEE Transactions on Neural Networks 18 (January (1)) (2007) 14–27.