Breast Fine Needle Tumor Classification using Neural Networks

Yasmeen M. George[*1], Bassant Mohamed Elbagoury, Hala H. Zayed[*3] and Mohamed I. Roushdy[#4]

[*]Computer Science Department, Faculty of computer and informatics,
Benha University, Qalyubiyah, Egypt

[#]Computer Science Department, Faculty of computer and information sciences,
Ain Shams University, Cairo, Egypt

Abstract

The purpose of this paper is to develop an intelligent diagnosis system for breast cancer classification. Artificial Neural Networks and Support Vector Machines were being developed to classify the benign and malignant of breast tumor in fine needle aspiration cytology. First the features were extracted from 92 FNAC image. Then these features were presented to several neural network architectures to investigate the most suitable network model for classifying the tumor effectively. Four classification models were used namely multilayer perceptron (MLP) using back-propagation algorithm, probabilistic neural networks (PNN), learning vector quantization (LVQ) and support vector machine (SVM). The classification results were obtained using 10-fold cross validation. The performance of the networks was compared based on resulted error rate, correct rate, sensitivity and specificity. The method was evaluated using six different datasets including four datasets related to our work and two other benchmark datasets for comparison. The optimum network for classification of breast cancer cells was found using probabilistic neural networks. This is followed in order by support vector machine, learning vector quantization and multilayer perceptron. The results showed that the predictive ability of probabilistic neural networks and support vector machine are stronger than the others in all evaluated datasets.

Keywords: *fine needle aspiration cytology (FNAC), learning vector quantization (LVQ), multi layer perceptron (MLP), probabilistic neural networks (PNN) and support vector machine (SVM).*

1. Introduction

Cancer is a big threat to human life. Based on statistics from the World Health Organization (WHO), deaths caused by cancer will reach about 12 million people in 2030. Thus, it has become a big challenge to fight against cancers from both medical practice and scientific research field. Cancers in their early stages are vulnerable to treatment while cancers in their most advanced stages are usually almost impossible to treat.

Palpable breast lesions can be accurately diagnosed by preoperative tests (like physical examination, mammography, fine-needle aspiration cytology, and core needle biopsy) [1,2]. Mammography is most often used for screening purposes rather than for precise diagnosis. It allows a physician to find possible locations of micro-calcifications and other indicators in breast tissue. When a suspicious region is found, the patient is sent to a pathologist for a more precise diagnosis. This is when the FNA is taken. Fine needle aspiration provides a way to examine a small sample of the questionable breast tissue that allows the pathologist to describe the type of the cancer in detail. It has gained popularity due to its fast and easy approach, being inexpensive [3,4].

In the literature many researchers have carried out intelligent diagnostic systems specifically to provide ‘second opinion’ for pathologists in making diagnosis [5-7]. One can find approaches to breast cancer classification: k-nearest neighbors, support vector machines, multilayered perceptron, radial basis network, general regression neural network and probabilistic neural network. Some of the mentioned approaches are concentrated on classifying FNA slides based on existing benchmark dataset [8]. Other techniques involve images containing isolated cells in the diagnosis system [9-11] which is not applicable in real life. The classification techniques are based on a number of cell features for the characterization of a cell as normal or abnormal [12,13]. But some techniques provide a small number of features which affects the classification results [14]. In this study the cell features are extracted from our dataset images. Moreover, the images contain overlapped cells and cell clusters. Also the slides are classified using different classification networks to find the optimum classification model. Also the classifiers are evaluated using six different datasets to find a general classification model for all these datasets.

In this paper, the studied dataset was based on microscopic images of breast Fine Needle Aspiration Cytology (FNAC) specimens obtained in cooperation with specialists from the archive of Early Cancer Detection Unit-Obstetrics and Gynecology Department, Ain shams University Hospitals. Samples were taken from breast lumps using 23-22G needle and spread on glass slides, stained with May Grunwald Giemsa stain or Diff. Quick stain. The dataset consists of 92 FNAC images, including 45 images of benign tumors, 47 of malignant tumors. The images were acquired through Olympus digital camera adapted to a trinocular optical microscope. Images were captured using 10x & 40x magnification lens. The size of the acquired images was 2560 x 1920. The images were stored in JPG format. The image itself was coded using the RGB color space and was not subject to any kind of lossy compression.

In our work, we present an intelligent classification system for breast microscopic cellular image. At the first phase, we extend our previous work [15] for the segmentation of nuclei boundaries with the determination of meaningful features for the detected cell areas. The nuclei boundaries are extracted based Hough transform in conjunction with watershed algorithm. Features describing texture and shape are calculated for each segmented region. These features contribute to the identification of the normal and abnormal cells. A clustering step is then performed for the classification of benign and malignant nuclei. Since artificial neural network gives fast and accurate classification and is a promising tool for tumor diagnosis. We have performed four intelligent classifiers namely multilayer perceptron using back-propagation algorithm, probabilistic neural networks, learning vector quantization and support vector machine.

The rest of the paper is divided into four sections. Section 2 describes the process of feature extraction phase used in our approach. Section 3 presents the four classifier models used in the proposed method. Section 4 shows the training and testing related to our work. Section 5 shows the experimental results and discussion. The last part of the work includes a conclusion and references. These sections are described in detail in the following paragraphs.

2. Feature Extraction

The efficient classification of nuclei cells from the total segmented regions requires the generation of meaningful features of very good discriminative ability. Having found the areas of the nuclei enclosed by the detected boundaries, features concerning the shape and the texture of the detected regions can be easily determined [16]. In our work, we use ten shape-based features and two textural features. The values obtained for these features yield a good differentiation between cancerous and healthy cells. These features are proposed as input data for the classification phase. The extracted features consist of:

Shape features. The detected boundaries for the nuclei are expected to present an ellipse-like shape and several features to describe this characteristic are chosen. Ten features are calculated from the extracted shape of the detected region boundary namely perimeter, compactness, smoothness, eccentricity, solidity, equivalent diameter, extent, major axis length and minor axis length. Table 1 shows detailed explanation for these features.

Textural features. Two features are calculated from the texture of the cell nucleus that is the standard deviation for the intensities of the region in both grayscale and Y-level.

3. Classification

Classification is a task of assigning an item to a certain category, called a class, based on the characteristic features of that item. This task in any classification system is performed by a classifier that takes a feature vector as an input and responds with a category to which the object belongs. A feature vector is a set of features extracted from the input data. In our study the feature vector represents the twelve features extracted for each nucleus as illustrated above in the feature extraction phase. Here we make use of neural network classifiers that are a collection of neurons (systems with many inputs and one output that are trained to fire, or not, for particular input patterns) that are connected one to another. Each connection is assigned an initial weight during the training process which is then adjusted to give a proper answer. The final decision is made based on the interaction of weights and the feature vector. The classification step was realized using four well known supervised classification algorithms: support vector machine, learning vector quantization, probabilistic neural networks and multilayer perceptron using back-propagation algorithm.

3.1 Support Vector Machine

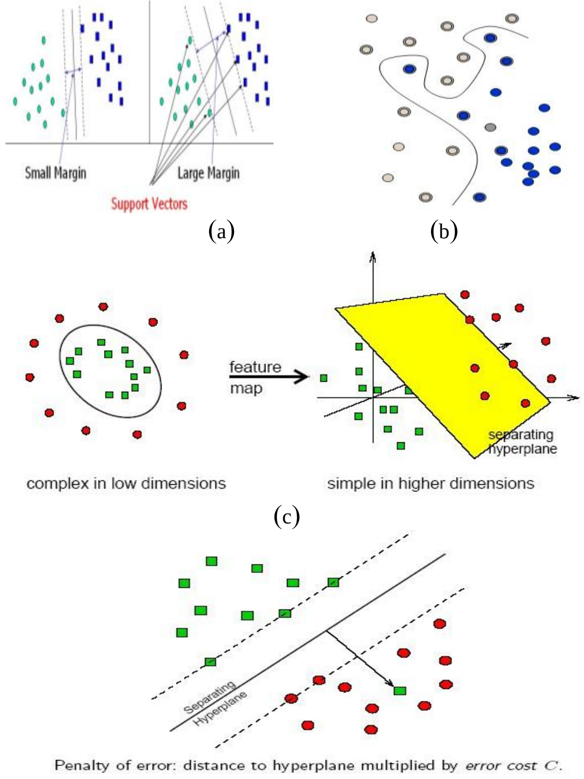
A Support Vector Machine (SVM) performs classification by constructing an N-dimensional hyperplane that optimally separates the data into two categories [17]. SVM models are a close cousin to classical multilayer perceptron neural networks. Using a kernel function, SVM’s are an alternative training method for polynomial, radial basis function and multi-layer perceptron classifiers in which the weights of the network are found by solving a quadratic programming problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in standard neural network training.

Table 1 Shape features used in this study

Feature	Definition
Perimeter	This is the total number of the nucleus boundary points.
Area	This is measured simply by counting the number of points inside the nucleus region.
Compactness	Perimeter and area are combined to give a measure of the cell nucleus compactness using the formula $C = 4\pi\ area/perimeter^2$. Compactness measures the efficiency with which a boundary encloses an area. For a circular region we have that C ≈ 1. This represents the maximum compactness value.
Smoothness	The smoothness of a nuclear contour is quantified by measuring the difference between the length of a radial line and the mean length of the lines surrounding it.
Eccentricity	Eccentricity specifies the eccentricity of the ellipse that has the same second-moments as the region. It allows us to track how much a segmented nucleus differs from a healthy nucleus. Healthy nuclei will assume circular shapes while cancerous nuclei can assume arbitrary shapes. We calculate eccentricity as the ratio of the distance between the foci of an ellipse and its major axis length. The values of this feature vary between 0 and 1. These are degenerate cases because an ellipse whose eccentricity is 0 is actually a circle, while a shape whose eccentricity is 1 is a line segment.
Solidity	Solidity specifies the proportion of the pixels in the convex hull that are also in the region. Computed as $Area/Convexarea$.
Equivalent Diameter	This specifies the diameter of a circle with the same area as the region. Computed as $\sqrt{4\ area\ /\ \pi}$.
Extent	Extent specifies the ratio of pixels in the region to pixels in the total bounding box. Computed as the Area divided by the area of the bounding box.
Major Axis Length	This specifies the length (in pixels) of the major axis of the ellipse that has the same normalized second central moments as the segmented nucleus.
Minor Axis Length	This specifies the length (in pixels) of the minor axis of the ellipse that has the same normalized second central moments as the segmented nucleus.

The goal of SVM modeling is to find the optimal hyperplane that separates clusters of vector in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other side of the plane. The vectors near the hyperplane are called the support vectors. An SVM analysis finds the hyperplane that is oriented so that the margin between the support vectors is maximized. As shown in Figure 1(a), the line in the right panel is superior to the line in the left panel. In general, for the data vectors with N dimensions, SVM analysis finds (N-1) dimensional hyperplane that separates them. Sometimes separating the clusters is achieved with non-linear curves Figure 1(b), SVM handles this by using a kernel function to map the data into a different space where a hyperplane can be used to do the separation Figure 1(c). The kernel function transforms the data into a higher dimensional space to make it possible to perform the separation. The concept of a kernel mapping function is very powerful. It allows SVM models to perform separations even with very complex boundaries.

Ideally an SVM analysis should produce a hyperplane that completely separates the feature vectors into two non-overlapping groups. However, perfect separation may not be possible. To allow some flexibility in separating the categories, SVM models have a cost parameter, C, that controls the tradeoff between allowing training errors and forcing rigid margins as shown in Figure 1(d). It creates a soft margin that permits some misclassifications. Increasing the value of C increases the cost of misclassifying points and forces the creation of a more accurate model that may not generalize well.


Small Margin
Large Margin
Support Vectors
feature map
separating hyperplane
complex in low dimensions
simple in higher dimensions
Separating Hyperplane
Penalty of error: distance to hyperplane multiplied by *error cost C*.

(a) (b)

(c)

(d)

Figure 1 (a) The best separable line with the maximum margin. (b) Separating the clusters with non-linear curves. (c) Separation may be easier in higher dimensions using kernel functions. (d) Non-separable training sets – use linear separation, but admit training errors.

3.2 Multi Layer Perceptron

Rumelhart, Hinton, and Williams (1986) revived interest in neural networks by introducing the generalized delta rule for learning by back propagation [18], which is today the most commonly used training algorithm for multilayer networks. Feed forward networks consist of a series of layers. The first layer has a connection from the network input. Each subsequent layer has a connection from the previous layer. The final layer produces the network's output. Feed forward networks can be used for any kind of input to output mapping. Each network input-to-unit and unit-to-unit connection is modified by a weight. In addition, each unit has an extra input that is assumed to have a constant value of one. The weight that modifies this extra input is called the bias. All data propagate along the connections in the direction from the network inputs to the network outputs. When the network is run, each hidden layer unit performs the calculation in Eq. 1 on its inputs and transfers the result (Oc) to the next layer of units. While each output layer unit performs the calculation in Eq. 2 on its inputs and transfers the result (Oc) to a network output. O_c is the output of the current layer unit c, P is either the number of units in the previous hidden layer or number of network inputs, $i_{c,p}$ is an input to unit c from either the previous hidden layer unit p or network input p, $w_{c,p}$ is the weight modifying the connection from either unit p to unit c or from input p to unit c, and b_c is the bias. $h_{Hidden}(x)$ is the sigmoid activation function for hidden units and $h_{Output}(x)$ is the linear activation function for output units. Other types of activation functions exist, but these functions were performed in this research.

$$Q_c = h_{Hidden}\left(\sum_{p=1}^{P} i_{c,p} w_{c,p} + b_c\right) \text{ where } \quad h_{Hidden}(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$Q_c = h_{Output}\left(\sum_{p=1}^{P} i_{c,p} w_{c,p} + b_c\right) \text{ where } \quad h_{Output}(x) = x \tag{2}$$

The neural network has to be trained on an appropriate data series to make meaningful forecasts. We use back-propagation training algorithm for this purpose. back-propagation training [18] consists of three steps:

1. Present an example's input vector to the network inputs and run the network: compute activation functions sequentially forward from the first hidden layer to the output layer.
2. Compute the difference between the desired output for that example, output, and the actual network output (output of unit(s) in the output layer). Propagate the error sequentially backward from the output layer to the first hidden layer.
3. An error term is computed for each unit in the output layer according to Eq. 3.

D_c is the desired network output (from the output vector) corresponding to the current output layer unit, O_c is the actual network output corresponding to the current output layer unit, and $h'_{Output}(x)$ is the derivative of the output unit linear activation function i.e 1. Also the error term is computed for each unit in the hidden layers according to Eq. 4. N is the number of units in the next layer (either another hidden layer or the output layer), δ_n is the error term for a unit in the next layer, and $w_{n,c}$ is the weight modifying the connection from unit c to unit n.

$$\delta_c = h'_{Output}(x)(D_c - O_c) \tag{3}$$

$$\delta_c = h'_{Hidden}(x) \sum_{n=1}^{N} \delta_n w_{n,c} \text{ where } \quad h'_{Hidden}(x) = O_k(1 - O_k) \tag{4}$$

$$\Delta w_{c,p} = \alpha \delta_c O_p \tag{5}$$

For every connection, change the weight modifying that connection in proportion to the error according to Eq. 5 where $w_{c,p}$ is the weight modifying the connection from unit p to unit c , α is the learning rate which controls how quickly and how finely the network converges to a particular solution, and O_p is the output of unit p or the network input p.

When these three steps have been performed for every example from the data series, one epoch has occurred. Training usually lasts until a predetermined maximum number of epochs (epochs limit) is reached or the network output error (error limit) falls below an acceptable threshold. Training can be time-consuming, depending on the network size, number of examples, epochs limit, and error limit.

3.3 Learning Vector Quantization

Having N data vectors, VQ algorithm groups them into small number of clusters in an unsupervised approach. VQ may be considered as a clustering process. However, LVQ neural network is a supervised classifier first introduced by by Kohonen's [19]. It combines clustering and classification processes based on feed forward neural network. The architecture of Kohonen's neural network that implements LVQ operations is shown in Figure 2. It consists of three layers; named input, hidden called competitive, and output called linear layers. Each neuron in the competitive layer represents one cluster. The linear layer maps the competitive layer's neurons into target classification defined by the user. Multiple neurons may belong to the same class, however, in the data space; cluster regions corresponding to the same class need not be contiguous.

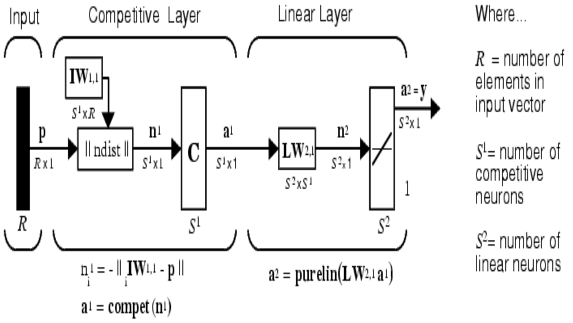


Figure 2 the architecture of kohonen's neural network that implements LVQ operations

Design and learning of LVQ

The first step in the design of LVQ neural network is setting the parameters of both competitive and linear layers. Then the available input data vectors have to be partitioned into training and test groups. Learning algorithm generally works as follows:

Codebook initialization: The number of codebook vectors for each target class has to be comparative to the number of occurrence of that class and these vectors are initialized to the center of the input ranges.

Winner determination: The Euclidean distance has to be calculated between training data vector and each codebook vector according to Eq. 6, where x is a training data vector, w is codebook vector, w is M-dimensional, and d is the Euclidean distance. The neuron m_c with codebook vector that has the least Euclidean distance to a data vector x is considered a winner according to Eq. 7.

$$d = \|w_j - x_i\| = \sum_i (w_{ji} - x_i)^2 \qquad (6)$$

$$m_c = \arg\min_j d_j \qquad (7)$$

Codebook Adaptation: Codebook vectors are optimized during learning process. Different learning algorithms have been proposed. They are all iterative gradient methods. The purpose is to find the optimal codebook and avoid complex gradient calculations

3.4 Probabilistic Neural Networks

In 1990, Donald F. Specht proposed a method to formulate the weighted-neighbor method in the form of a neural network [20]. He called this a "Probabilistic Neural Network". Probabilistic neural networks are conceptually similar to K-Nearest Neighbor (k-NN) models. The basic idea is that a predicted target value of an item is likely to be about the same as other items that have close values of the predictor variables.

The nearest neighbor classification depends on how many neighboring points are considered for the classification of a new pattern. A probabilistic neural network builds on this foundation and generalizes it to consider all of the other patterns. The distance is computed from the pattern being evaluated to each of the other patterns, and a radial basis function (RBF) (also called a kernel function) is applied to the distance to compute the weight (influence) for each pattern. The radial basis function is so named because the radius distance is the argument to the function (Weight = RBF(distance)). The further some other pattern is from the new pattern, the less influence it has. The architecture for PNN is shown in Figure 3. As illustrated in the figure, PNN network has four layers:

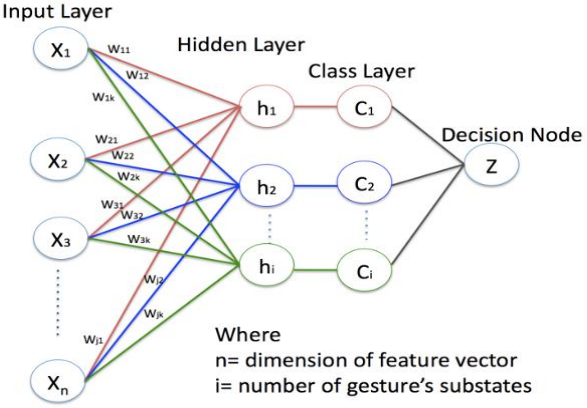


Figure 3 Probabilistic Neural Network Architecture

Input layer — there is one neuron in the input layer for each predictor variable. *N*-1 neurons are used where *N* is the number of categories. The input neuron (or processing before the input layer) standardizes the range of the values by subtracting the median and dividing by the inter-quartile range. The input neurons then feed the values to each of the neurons in the hidden layer.

Hidden layer — this layer has one neuron for each case in the training data set. The neuron stores the values of the predictor variables for the case along with the target value. When presented with the *x* vector of input values from the input layer, a hidden neuron computes the Euclidean distance of the test case from the neuron's center point and then applies the RBF kernel function using the sigma value as shown in Eq. 8. The resulting value is passed to the neurons in the pattern layer.

$$(hidden\_unit)\varphi_j = \exp(-\frac{\sum_{i=1}^{n}(x_i - w_{ij})^2}{2\sigma^2}) \qquad (8)$$

Pattern layer / Summation layer — there is one pattern neuron for each category of the target variable. The actual target category of each training case is stored with each hidden neuron; the weighted value coming out of a hidden neuron is fed only to the pattern neuron that corresponds to the hidden neuron's category. The pattern neurons add the values for the class they represent (hence, it is a weighted vote for that category).

Decision layer — the decision layer compares the weighted votes for each target category accumulated in the pattern layer and uses the largest vote to predict the target category.

4. Training and Testing

Training and testing datasets are constructed using 10-fold cross-validation [21]. The performance of the classifier is calculated using the unknown nuclei features of the testing dataset. Some researches provide one feature vector for each FNAC image by extracting N features for each nucleus and then calculate some measures for each feature for all nuclei in this image [7,10,14]. Wisconsin Diagnostic Breast Cancer is the most commonly used dataset by many researchers [8]. Wisconsin dataset computes three measures which are the mean, standard error $Se = (Std\_dev) / \sqrt{length}$, and "worst" or largest (mean of the three largest values). In this case the number of patterns will be equal to the number of data set images. Other researchers generate one feature vector for each nucleus [9,11]. This is performed when the number of dataset images is fairly small. The generated dataset features will contain number of patterns equals to the number of nuclei in all dataset images.

In our work, we have constructed four different input data matrices from our dataset images. As mentioned before, our dataset images contain 92 FNAC image. The first input data matrix (dataset 1) contains one feature vector for each FNAC image with dimension of 92 x 60. Each FNAC slide is represented by sixty feature, twelve features are extracted for each nucleus then five measures are computed for each feature. The five calculated measures in our work are min, max, mean, standard error, and "worst" (mean of the three largest values). The second input data matrix (dataset 2) is the same as dataset 1 but we apply filtering for removing large connected components in the segmented image before feature extraction. As we have found that many contiguous cells are extracted as one connected component as illustrated in Figure 4(a). The values of the features will be largely dependent on the size of the connected component. So we have filtered the segmented image by removing the connected component with area greater than some threshold. Figure 4(b) shows the segmented image after the removal of large components.

The third input data matrix (dataset 3) contains one feature vector for each nucleus with dimension of 3260 x 12. Each nucleus is represented by twelve features. 3260 is the number of connected components in all images in our dataset. Finally the fourth input data matrix (dataset 4) is the same as dataset 3 but after removing the large components. The dimension of dataset 4 is 2561 x 12. This means that we have removed 699 connected components.

We have trained the four constructed datasets. Also we have used two benchmark datasets with different features to compare the classification results with our datasets results. Breast Cancer Wisconsin (Diagnostic) dataset [22] (dataset 5) with dimension of 569 x 30 where 569 represent the number of dataset images and 30 is the feature vector length. The last trained dataset is Breast Cancer Wisconsin (Original) dataset [23] (dataset 6).

Table 2 shows detailed description about all six datasets. We have trained each dataset using the four illustrated classifiers.

The parameters needed for training and testing the datasets based on the four classifiers are shown in Table 3. We have got the values of these parameters after several experiments. The training and testing sets were chosen using 10-fold cross-validation. We took into consideration four neural networks classifiers: support vector machines (SVM), multilayer perceptron (MLP), learning vector quantization (LVQ) and probabilistic neural networks (PNN). We applied these classifiers on the six datasets that contain our study datasets and benchmark datasets as illustrated above. The training times for the classifiers presented in this study are provided in Table 4. As shown in the table, the PNN classifier takes the lowest training time. This is followed by SVM, LVQ and MLP. For the training results, Figure 5 shows the LVQ training receiver operating characteristic curve (ROC) for the six datasets. Also Figure 6 shows the MLP training performance for the six datasets.

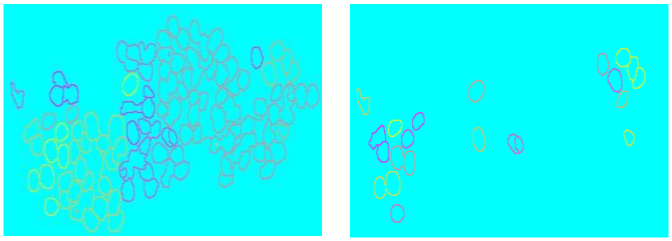
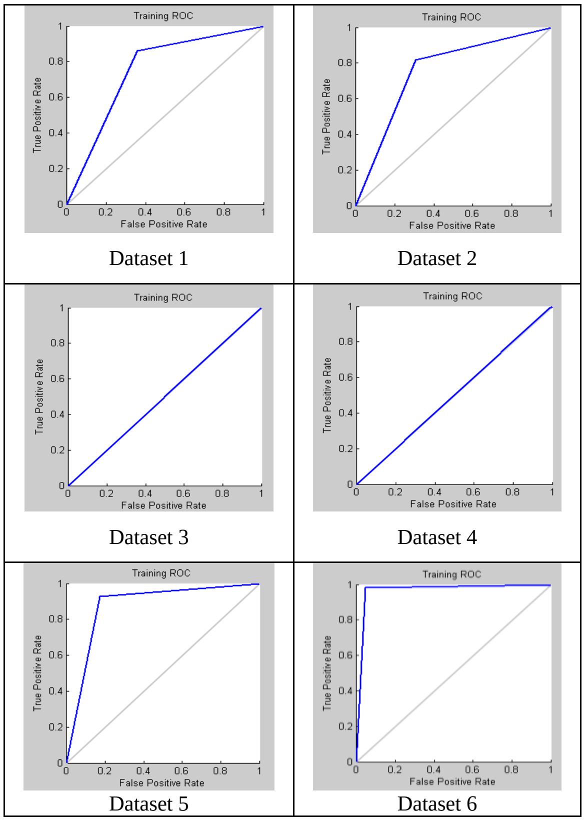

(a) (b)

Figure 4 (a) Cell Nuclei before removing large connected components. (b) Cell Nuclei after removing large connected components

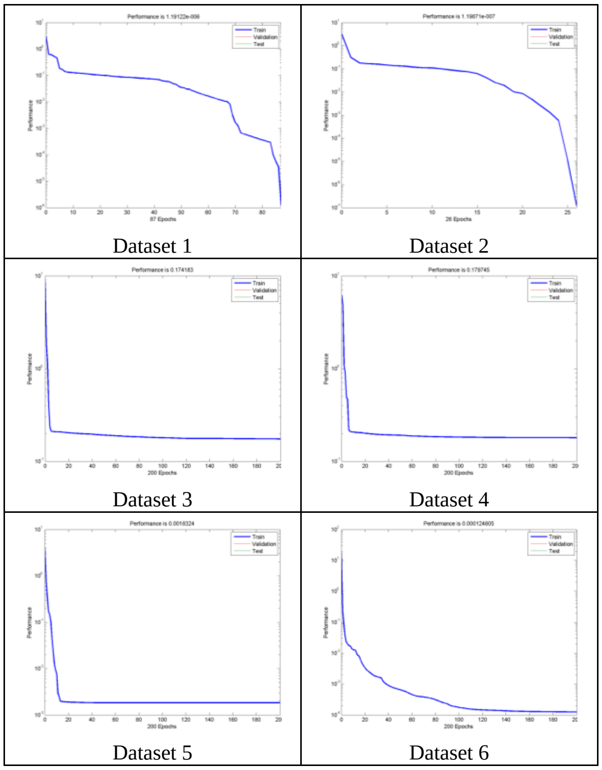
5. Results and Discussion

We will demonstrate the performance of the four classifiers presented in Section 2 along with results obtained for applying these classifiers on the six illustrated datasets for comparison. The method was developed in Matlab version 10b using a dual core PC with a 2.0 GHz processor and 2 GB of RAM.



Dataset 1 Dataset 2 Dataset 3 Dataset 4 Dataset 5 Dataset 6

Figure 5 LVQ Training ROC for six datasets



Dataset 1 Dataset 2 Dataset 3 Dataset 4 Dataset 5 Dataset 6

Figure 6 MLP Training Performance for six datasets

Table 2 Details of the datasets used in training and testing

	No. of patterns	Features	Description
Dataset 1	92 Image 45 Benign 47 Malignant	60 features: five measures (min , max, mean, standard error and worst) and twelve features (Perimeter, Area, Compactness, Smoothness, Eccentricity, Solidity, Equivalent Diameter, Extent, Major Axis Length, Minor Axis Length, Standard deviation for grayscale intensities and Standard deviation for Y-level intensities).	Nuclei statistical features Before large components removing Feature vector for each image
Dataset 2	92 Image 45 Benign 47 Malignant	Same as dataset 1.	Nuclei statistical features After large components removing Feature vector for each nucleus
Dataset 3	3260 Image 1135 Benign 2125 Malignant	Twelve features (Perimeter, Area, Compactness, Smoothness, Eccentricity, Solidity, Equivalent Diameter, Extent, Major Axis Length, Minor Axis Length, Standard deviation for grayscale intensities and Standard deviation for Y-level intensities).	Nuclei statistical features Before large components removing Feature vector for each image
Dataset 4	2561 Image 933 Benign 1628 Malignant	Same as dataset 3.	Nuclei statistical features After large components removing Feature vector for each nucleus
Dataset 5	569 Image 357 Benign 212 Malignant	30 features: three measures (mean, standard error and worst) and ten features (radius, texture, standard deviation of gray-scale values, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension).	Nuclei statistical features Feature vector for each image

Dataset 6	699 Image 458 Benign 241 Malignant	9 features: Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli	Breast tissue features Feature vector for each image

Table 3 Training parameters used in SVM, LVQ, PNN and MLP

SVM	Number of iterations =1000
	Kernel function : Linear
	C parameter = 1
MLP	Number of hidden layers = 1
	Hidden layer neurons = 24
	Hidden activation function = sigmoid function
	Output layer neurons = 2
	Output activation function = linear function
	Max number of iterations = 200
	Goal (error limit) = 1e-05
	Learning algorithm : Levenberg-Marquardt back-propagation
LVQ	Number of hidden neurons =30
	Learning rate =0.01
	Max number of iterations = 100
	Goal (error limit) = 1e-04
	Learning function = 'learnlv1'
PNN	Goal (error limit) = 1e-05

Two widely used statistical measures for the performance of the classification are calculated; sensitivity, specificity. The sensitivity measures the proportion of malignant tumors which are correctly identified; it is calculated as Eq. 9. The specificity measures the proportion of benign tumors that are correctly characterized; it is calculated as Eq. 10. We have used these measures for the performance analysis and for providing tools to select a possibly optimal classification model.

$$\text{sensitivity} = \frac{TP}{TP + FN} \qquad (9)$$

$$\text{specificity} = \frac{TN}{TN + FP} \qquad (10)$$

Table **5** shows the sensitivity measure of the four tested classifiers for the six datasets and Table 6 shows the specificity measure. First, both tables show that the sensitivity and specificity measures for dataset 1 are better than dataset 2 and also both measures for dataset 3 are better than dataset 4 for all classifiers. As mentioned before, dataset 1 and dataset 3 contain features for all connected components while dataset 2 and dataset 4 contain features for connected components after removing the large ones. This means that removing large connected components from the segmented nuclei does not improve any classifier performance. As all classifiers gives higher sensitivity and specificity in datasets with features without removing large connected components (dataset 1 and dataset 3).

Next, both tables show that the sensitivity and specificity measures for dataset 1 and dataset 2 are lower than dataset 3 and dataset 4 for all classifiers. As mentioned before, dataset 1 and dataset 2 include one feature vector for each FNAC image while dataset 3 and dataset 4 include one feature vector for each nucleus. This means that extracting feature vector for each nucleus is better than extracting feature vector for each FNAC image.

Next, the average measures for each classifier across all six datasets show that the best classifier is PNN. This is followed in order by SVM, LVQ and MLP. As shown in the below tables, PNN classifier gives the best measures for four datasets (dataset 1, dataset 2, dataset 5 and dataset 6) while SVM gives the best measures for two datasets (dataset 3 and dataset 4). It must be noted that PNN and SVM take very small training times when comparing with LVQ and MLP training times as shown in Table **4**.

Finally, we have trained the benchmark Wisconsin datasets (dataset 5 and dataset 6) for the comparison with our study datasets. The results show that SVM and LVQ networks give the highest sensitivity and specificity with our study datasets (dataset 3 and dataset 4) while PNN and MLP networks give the highest measures with the Wisconsin datasets (dataset 5 and dataset 6). This means that our dataset features are accurate and can be compared with benchmark dataset features.

Figure 7 includes a chart for the error rates and correct rates of the tested classifiers for each training dataset. All the achieved results show that the predictive ability of both probabilistic neural network and support vector machine are stronger than the learning vector quantization and multilayer perceptron for the evaluated datasets.

Table 4 Training times for the classifiers used in this study

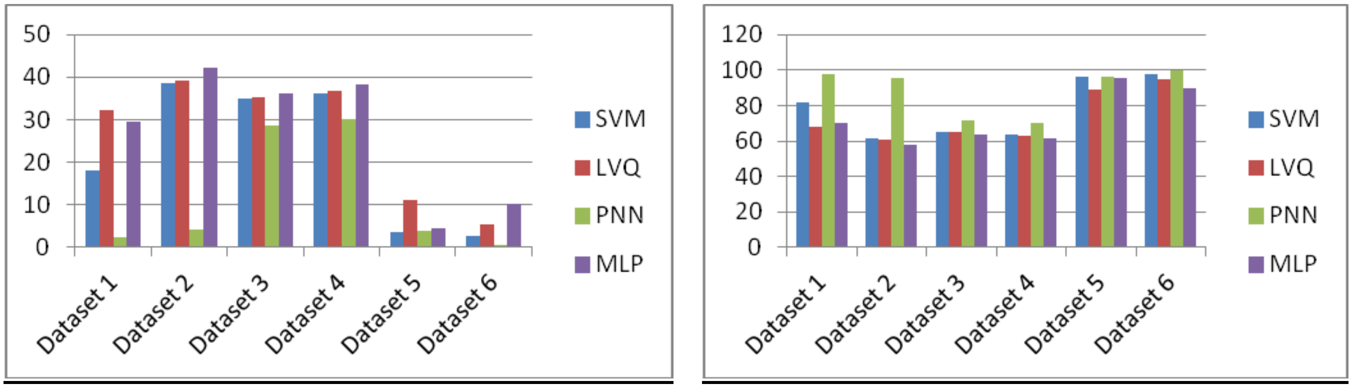
Classifiers	**Dataset 1**	**Dataset 2**	**Dataset 3**	**Dataset 4**	**Dataset 5**	**Dataset 6**
	Time in sec (mean ± std)					
SVM	0.44 ± 0.98	0.09 ± 0.01	41.4 ± 6.61	33.54 ± 10.18	47.19 ± 2.44	87.77 ± 2.49
LVQ	71.84 ± 3.97	69.6 ± 0.66	2145.22 ± 21.16	1681.82 ± 25.64	387.95 ± 7.86	463.45 ± 4.37
PNN	0.43 ± 0.8	0.18 ± 0.0	0.19 ± 0.0	0.18 ± 0.0	0.18 ± 0.0	0.18 ± 0.0
MLP	133.65 ± 52.42	30.17 ± 12.97	115.8 ± 6.89	95.52 ± 2.06	144.39 ± 53.43	19.56 ± 6.8

Table 5 Sensitivity of the tested classifiers for the six different datasets

Classifier	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Average sensitivity
SVM	78.82	59.44	99.6	98.12	98.77	96.1	88.48
LVQ	83.23	58.6	98.86	98.01	92.34	96.65	87.95
PNN	97.02	94.84	93.78	92.48	100	99.8	96.32
MLP	74.85	57.7	81.3	75.41	97.93	96.74	80.66

Table 6 Specificity of the tested classifiers for the six different datasets

Classifier	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Average specificity
SVM	92.23	64.09	99.64	98.65	94.63	96.68	90.99
LVQ	79.44	60.99	99.03	97.76	84.23	92.69	85.69
PNN	97.28	95.17	93.67	92.22	89.87	99.23	94.57
MLP	68.86	57.23	82.46	77	94.94	77.19	76.28



)a( )b(

Figure 7 (a) Error rates of the tested classifiers for each training dataset. (b) Correct rates of the tested classifiers for each training dataset

6. Conclusion and Future Work

FNAC is an essential component in the preoperative management of breast lesions. Its accuracy, ease of use, and affordability are factors that cause its popularity. The advent of imaging technology together with the clinical expertise of the clinician contributed to its increased sensitivity.

In this paper, we have developed a computer-aided diagnosis system for breast FNAC classification. For the feature extraction, we used ten shape-based features and two textural features. The values obtained for these features yield a good differentiation between benign and malignant cells. For the classification phase, we have performed four different classification models namely multilayer perceptron, probabilistic neural networks, learning vector quantization and support vector machine. The classification results were obtained using 10-fold cross validation.

Six datasets were used to examine the efficiency of the proposed classifiers. Four datasets were constructed from our material and the other two datasets are benchmark datasets and were evaluated for the comparison. The conducted experiments show that the classification results of the datasets contain one feature vector for each nucleus with features for all connected components without removing large ones are the best datasets for training based on all proposed classifiers.

The classification performance was capable of producing up to 99.7 % sensitivity and specificity for our datasets. The results showed that the predictive ability of both probabilistic neural network and support vector machine was stronger than the learning vector quantization and multilayer perceptron for the evaluated datasets.

The results obtained so far are encouraging; more investigations are needed to further improve the clustering algorithms results (FCM, SVM, LVQ, MLP and PNN) with the selection of different nuclei features and performing hybrid clustering algorithms. Also the implementation of other clustering algorithms needs to be investigated. Furthermore, the proposed diagnostic system is being tested in a larger dataset in order to evaluate the robustness of this system. More work need to be done in the classification of cell nuclei malignancy to know the degree of malignancy of the FNAC image. Finally, the proposed diagnosis system can be used as the basis for further applications, such as mobile application for remote diagnosis for breast cancer.

7. Acknowledgement

The authors would like to thank Prof. Zeinab Shehab El-Din, Professor of Pathology, Early Cancer Detection Unit-Obstetrics and Gynecology Department, Faculty of Medicine; Ain shams University for providing FNA breast cytological slides for the construction of the image dataset.

References

[1] Hamaguchi Y, Tanabe M, Momiyama N, Chishima T, Nakatani Y, Nozawa A, Sasaki T, Kitamura H, Shimada H. Ishikawa T, "False-positive and false-negative cases of fine-needle aspiration cytology for palpable breast lesions," Breast Cancer, vol. 14, no. 4, pp. 388–392, 2007.

[2] Paulo Mendoza, Maribel Lacambra, Puay-Hoon Tan, and Gary M. Tse, "Fine Needle Aspiration Cytology of the Breast: The Nonmalignant Categories," Pathology Research International, vol. 2011, p. 8, 2011.

[3] Manon Auger and Istvan Huttner, "Fine-needle aspiration cytology of pleomorphic lobular carcinoma of the breast: Comparison with the classic type," vol. 81, no. 1, pp. 29-32, February 1997.

[4] Mulazim H. Bukhari and Zahid M. Akhtar, "Comparison of accuracy of diagnostic modalities for evaluation of breast cancer with review of literature," Diagnostic Cytopathology, vol. 37, no. 6, pp. 416-424, 2009.

[5] Mahjabeen Mirza Beg and Monika Jain, "an Analysis of the methods employed for Breast Cancer Diagnosis," International Journal of Research in Computer Science, vol. 2, no. 3, pp. 25-29, 2012.

[6] Nuryanti Salleh, Mohd Arshad, Nor Othman Harsa Sakim, "Evaluation of Morphological Features for Breast Cells Classification Using Neural Networks," in Tools and Applications with Artificial Intelligence.: Springer, 2009, vol. 166, pp. 1-9.

[7] Isa, N.A.M., E. Subramaniam, M.Y. Mashor, and N.H. Othman, "Fine Needle Aspiration Cytology Evaluation for Classifying Breast Cancer Using Artificial Neural Network," American Journal of Applied Sciences, vol. 4, no. 12, pp. 999-1008, 2007.

[8] Tüba Kiyan and Tülay Yildirim, "Breast Cancer Diagnosis Using Statistical Neural Networks," Journal of Electrical & Electronics Engineering, vol. 4, 2004.

[9] Marek Kowal, Paweł Filipczuk, Andrzej Obuchowicz, and Józef Korbicz, "Computer-Aided Diagnosis Of Breast Cancer Using Gaussian Mixture Cytological Image Segmentation," Journal Of Medical Informatics & Technologies, vol. 17, 2011.

[10] Shekhar Singh et al., "Breast Cancer Detection and Classification of Histopathological Images," International Journal of Engineering Science and Technology (IJEST), vol. 3, no. 0975-5462, p. 5, May 2011.

[11] William N. Street, "Xcyt: A system for remote cytological diagnosis and prognosis of breast cancer," Soft Computing Techniques in Breast Cancer Prognosis and Diagnosis, pp. 297–322, 2000.

[12] P. Filipczuk, M. Kowal, and A. Marciniak, "Feature selection for breast cancer malignancy classification problem," Journal of Medical Informatics & Technologies , vol. 15, pp. 193-199, 2010.

[13] R. NITHYA and B. SANTHI, "Comparative Study On Feature Extraction Method For Breast Cancer Classification," Journal of Theoretical and Applied Information Technology, vol. 33, pp. 220 - 226, November 2011.

[14] Lukasz Jelen, Thomas Fevens, and Adam Krzyzak, "Classification of Breast Cancer Malignancy Using Cytological Images of Fine Needle Aspiration Biopsies," International Journal of Applied Mathematics and Computer Science, vol. 18, no. 1, pp. 75-83, March 2008.

[15] Y.M. George, B. M. Bagoury, H. H. Zayed, M. I. Roushdy, "Automated Cell Nuclei Segmentation for Breast Fine Needle Aspiration Cytology," Breast Cancer Special Issue in Signal Processing Journal (2012), http://dx.doi.org/10.1016/j.sigpro.2012.07.034.

[16] William H. Wolberg, W.Nick Street, and O.L. Mangasarian, "Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates," Computer applications for early detection and staging of cancer, vol. 77, no. 2-3, pp. 163–171, March 1994.

[17] Nello Cristianini and John Shawe-Taylor, An Introduction to Support Vector Machines and other kernel-based learning methods.: Cambridge University Press, 2000.

[18] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing 2nd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[19] Kohonan T., "The self-organizing map," Proceedings of the IEEE, vol. 78, no. 9, pp. 1464-1480, 1990.

[20] Donald F. Specht, "Probabilistic Neural Networks," Neural Networks, vol. 3, pp. 109-118, 1990.

[21] Seymour Geisser, Predictive Inference, 1st ed. New York: Chapman and Hall, 1993.

[22] William H. Wolberg. Breast Cancer Wisconsin (Diagnostic) Dataset. [January, 2011]. http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

[23] WIlliam H. Wolberg. Breast Cancer Wisconsin (Original) dataset. [January, 2011]. http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29