Comparative evaluation of pattern recognition techniques for detection of microcalcifications

Kevin S. Woods[1], Jeffrey L. Solka[2], Carey E.Priebe[2], Christopher C. Doss[3], Kevin W. Bowyer[3], Laurence P. Clarke[1]

1. Center for Engineering and Medical Image Analysis (CEMIA), University Technology Center, USF, Tampa, FL 33620
2. Systems Research & Technology Dept, Advanced Computation Technology,B10,Naval Surface Warfare Center, Dahlgren, VA 22448-5000
3. Department of Computer Science, University of South Florida, Tampa, FL 33620

ABSTRACT

Computer detection of microcalcifications in mammographic images will likely require a multi-stage algorithm that includes segmentation of possible microcalcifications, pattern recognition techniques to classify the segmented objects, a method to determine if a cluster of calcifications exists, and possibly a method to determine the probability of malignancy. This paper will focus on the classification of segmented objects as being either (1) microcalcifications or (2) non-microcalcifications. Six classifiers (2 Bayesian, 2 dynamic neural networks, a standard backpropagation network, and a K-nearest neighbor) are compared. Methods of segmentation and feature selection are described, although they are not the primary concern of this paper. A database of digitized film mammograms is used for training and testing. Detection accuracy is compared accross the six methods.

I. INTRODUCTION

Radiologists look for certain signs and characteristics indicative of cancer when evaluating a mammogram. Among these signs is the presence of clustered microcalcifications. A microcalcification is a tiny calcium deposit that has accumulated in tissue in the breast, and it appears as a small bright spot on the mammogram. A cluster is typically defined to be at least 3 microcalcifications within a $1cm^2$ region. Between 30% and 50% of breast cancers demonstrate clustered microcalcifications, and in approximately 36% of these cases the clusters are the only sign of malignancy[2]. The calcifications that may be of potential interest as an indication of malignancy vary in size from .1mm to 5mm, and a radiologist must carefully examine the mammogram with a magnifier to locate calcifications which may be embedded in dense parenchymal tissue. Due to their size and subtlety, individual microcalcifications can easily be overlooked in the normal manual examination of the mammograms.

This paper compares six classification techniques for use in automatic microcalcification detection in digitized mammograms. Two Bayesian classifiers will be examined. They are (1) the linear classifier (LC) and (2) the quadratic classifier (QC). Three artificial neural networks (ANN) will be examined. They are (1) a standard feed-forward network using backpropagation training (BP), (2) Cascade Correlation (CC) ANN, and (3) Divide and Conquer (DCN) ANN. The CC and DCN are dynamic, or self-organizing, ANNs. A self-organizing ANN is one which automatically determines its topology (architecture) during training. The sixth method of pattern classification will be the k-nearest neighbors algorithm (KNN). The same training data will be used for the knowledge acquisition stage in each method, and a common set of test data will be used to evaluate and compare the methods. The methods used to acquire the training and test data remain independent of the classifiers, therefore the principal purpose of this paper will be to determine which, if any, of the classifiers produces superior results. The classifiers are trained to detect *individual* microcalcifications. Since clusters of microcalcifications are a sign of a possible malignancy, the results of the individual microcalcification detection can be further processed to indicate the presence of clusters. It is reasonable to assume that the classifier which produces the best results for individual microcalcification detection will give among the best results for cluster detection.

2. RELATED WORK

Previous research has focused on the detection of clustered microcalcifications, and no method has produced clinically acceptable results[1,2,3,4,5,6,7,8,9,10,11]. Many different pattern recognition methods have been used for microcalcification detection, including k-nearest neighbors[4], thresholds on each computed feature[1,5,6,8,11], binary decision trees[4], and Bayesian classifiers[10] to name a few. Most of these algorithms involve a segmentation process followed by a classification process. The two most successful methods are reported by Chan and Doi[1] and Davies and Dance[8]. Davies and Dance used 25 training images and 50 test images, half of which contained clusters of microcalcifications. They report successfully detecting 47 of 49 clusters with a total of 9 false positive clusters detected. Chan and Doi tested their algorithm using simulated microcalcifications

0-8194-1138-8/93/$6.00

superimposed on normal mammograms, and on 20 hand selected clinical images. For the simulated microcalcifications, they report 80% TP cluster detection rate with one FP cluster per image. For the clinical study, they report an 82% TP cluster detection rate with one FP cluster per image. We have not found any paper which has reported results for an automated screening environment, simulated or otherwise.

3. PATTERN CLASSIFICATION TECHNIQUES

Pattern recognition techniques can be used to assign objects to one of a fixed number of classes by means of some measured properties or features. The features are usually organized into a one-dimensional feature vector of real values which have been normalized so that all features are weighted equally. The features vectors can be plotted in a d-dimensional feature space, where d is the number of elements (features) in the feature vector. The classification of an unknown pattern can be determined by its location in feature space. One method is to divide the feature space into regions, where each region corresponds to a different class. The regions are divided by decision boundaries, and it is the job of the classifier to determine where these boundaries should be located in order to provide the highest possible classification accuracy.

A complete pattern recognition system requires data acquisition, data representation, and classification. The data acquisition and representation stages will remain unchanged for the six methods of classification to be compared, and will be discussed later. In a paper by Jain[12], classifiers are grouped according to the following dichotomies: finite or infinite number of training samples, labelled or unlabelled samples, and known or unknown form of the class-conditional density functions. For the classification problem at hand, we have a finite number of labelled training samples for which the density functions are unknown. At this point, we can either estimate the density functions or use methods which avoid a direct density function estimation. We have chosen to examine both parametric density estimation approaches and also those techniques which do not directly estimate the density functions but rather the decision boundaries. The former case includes the linear and quadratic classifiers, while the later includes the KNN and the ANNs. The following subsections will give thumbnail sketches of the different classifiers.

3.1 Linear and Quadratic Classifiers

The linear and quadratic classifiers are special cases of the more general Bayesian classifier. Bayesian classifiers can be applied to the multi-class case, but our interests here lie in the two class problem. Given a set of observations drawn from each of two classes, c_0 and c_1, one first develops an estimate of the underlying probability distribution $P(x|c_i)$ for each of the two classes. This information can be used to compute the posterior probabilities of interest

$$P(c_j|x) = \frac{P(x|c_j)P(c_j)}{p(x)}$$

where $p(x) = \sum_{j=1}^{2} p(x|c_j)p(c_j)$ and $p(c_j)$ is the prior probability for class j.

Given the true posterior probabilities for the two classes, one can form a discriminant function using $g(x) = p(c_0|x) - p(c_1|x)$. Given g(x), we classify a point as class c_0 if $g(x)>0$. It is well known that this is an optimal classification scheme[25]. So our goal is to model the posterior probabilities as accurately as possible.

The classifier type is determined by the model which is used to build the conditional density functions, $p(x|c_i)$. In many applications, a feature is well modeled by a Gaussian distribution $N(x,m,\Sigma)$ given by

$$N(x,\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp((x-\mu)'\Sigma^{-1}(x-\mu))$$

where x is an n-component column vector, μ is the n-component mean vector, Σ is the n-by-n covariance matrix, $(x-\mu)^t$ is the transpose of $x-\mu$, Σ^{-1} is the inverse of Σ, and $|\Sigma|$ is the determinant of Σ.

In the simplest case, the mean of each class is estimated separately and the covariance matrix is computed in a pooled manner, i.e. by averaging the covariance matrices of the two classes. In this case, it can be shown that the classifier boundary that separates the two classes is a hyperplane[20], and hence the classifier is referred to as a linear classifier or linear machine. If each distribution is allowed to have its own covariance matrix, then the decision surfaces that arise are referred to as hyperquadrics and the classifier is referred to as a quadratic classifier. These decision surfaces can assume the forms of pairs of hyperplanes, hyperspheres, hyperellipsoids, hyperparaboloids, and hyperhyperboloids.

One advantage of the probability density estimation approach to classification is the ability to use the estimates of the $p(x|c_i)$ in such a manner that the probability α of misclassifying a calcification may be adjusted depending on the circumstances. For example, if we base our decision process on the ratio of the likelihood for the two classes $p(x|c_0)/p(x|c_1)$, we can use this information to set a decision threshold which matches our desired level of α[21]. First we use the LC or QC procedure to estimate the conditional density functions for a certain training set, then by examining the set of likelihood ratios for all the points in the training set we can set the threshold T such that we reach the desired alpha level. Then we evaluate the classifier's performance on the test set using this threshold. In this manner, we get an estimate of α and the false alarm rate that one would expect in practice.

Another advantage of this approach is that there is no requirement that each of the two classes be equally represented in the training sets. Unlike the neural network approaches that follow, there is no such requirement for balance in the training set. All of the approaches used require that the training and test set be representative of the data. So as not to bias the testing results in a negative manner, it is particularly important that the training data is a good representation of the testing data.

There is at least one advantage that the approaches that estimate just the discriminant boundaries have over those that estimate the full posterior probability density functions (pdf). It takes more data to estimate the pdfs than the discriminant boundaries. This problem trend is particularly evident in the higher dimensional spaces where the curse of dimensionality reigns. In those cases where there is a limited amount of high dimensional training data, the advantages that the pdf approaches present may be overwhelmed by this fact.

3.2 Neural networks in general

In general, neural networks are characterized by (a) large numbers of simple processing nodes, (b) larger numbers of weighted connections between nodes in which knowledge is encoded, (c) highly parallel, distributed control, and (d) an ability to learn the internal representation (i.e. values for interconnection weights) automatically[12]. The processing nodes perform a weighted sum of their inputs, for which a single output value is computed using some nonlinear activation function. The node is said to "fire" if the activation value is greater than some adjustable threshold associated with the node. Network topology refers to the way that the processing nodes interconnections are made. The knowledge of a network is encoded in the interconnection weights, and it is the learning algorithm which specifies the initial values of the weights and how they are to updated to improve performance. ANNs can be described by their node characteristics, network topology, and the learning algorithm used to train the network.

The ANNs described in this paper are feed-forward networks. In a fully connected network, each node in one layer is connected to every node in the next layer. The input nodes serve only to distribute the input values to other nodes in the network. When the network is to be used for pattern classification, there is one input node for each element of the feature vector which describes a pattern. There may be one or more layers of hidden nodes which perform the weighted summing of inputs and pass an activation value as an output. The hidden nodes are so-called because their outputs are not directly observable, and are used as inputs to other nodes. Finally, there is a layer of output nodes, and it is the activation values of these nodes which are taken to be the output of the network. There is usually one output node for each possible class to which a pattern may be assigned. In feed-forward ANNs, the inputs are presented at the input nodes and the activations of the nodes flow through the network towards the output nodes where they can be observed.

The hidden nodes enable internal representations of the input data to be developed by the network during learning. These hidden nodes act as high order "feature detectors" which will fire (pass a positive activation value) in the presence of a particular feature and, conversely, pass an inhibitory signal (negative activation value) if the feature is absent. The term high order feature refers to one which is inferred by the ANN from the input data, and is not to be confused with the features that are computed by a user and specified as input to the ANN. The firing threshold associated with each node may be implemented as a bias input to the node with a value of 1 and an adjustable weight. This way the threshold may be

represented as a separate weighted connection, and can be learned during training as if it were any other weight. When an unknown pattern is presented to a trained network, a set of hidden nodes will fire and excite one of the output nodes. The connections from other hidden nodes will pass inhibitory signals to the other outputs, and the pattern is assigned to the class which is represented by the output node that fires. More than one output node may fire, and in this case the node with the maximum activation value is selected.

Neural networks, when applied to pattern classification, attempt to define decision regions in a d-dimensional feature space, where d is the number of inputs. The decision regions are defined by the interconnection weights in the network. In principle, any arbitrarily complex decision region can be formed by a multi-layer ANN with two hidden layers[14] (with possibly an arbitrary number of hidden nodes). Programming the weights by hand to specify decision regions is impossible, so they must be learned from a set of training examples which are a set of input values with corresponding desired (target) output values. The basic approach to training a feed-forward ANN is to present the training sample inputs to the network, allow the activations to flow to the output nodes, compare the network outputs with a desired (target) output to compute an error measure, and then somehow adjust the weights so that the error will be reduced. This procedure is repeated until the network converges on a solution. The most common estimate for the error of a network which is used to update the connection weights is the sum of squared errors (observed value - target value) over all outputs. Depending on the learning algorithm the error estimate may be computed for individual training samples or for the entire training set. An epoch is defined as one pass of all training samples through the network. Different learning algorithms specify if the network weights are to be updated after each training sample (for which the error estimate is computed for the training sample just presented), or after each epoch (in which case the error estimate is computed over the entire training set), and how they are to be updated.

3.3 Simple backpropagation neural networks

The first type of ANN considered is a fully connected backpropagation network (BP) with two hidden layers. Each node has a bias input of 1 with an adjustable weight. The activation function for hidden and output nodes is a sigmoid function which produces a real value between 0 and 1 based on the weighted sum of inputs. The major problem with multilayer networks is the training. The weights to the output nodes can be adjusted easily because there are target outputs. An error estimate is made for each output, and the weights connected to the output unit can be adjusted in a gradient descent method which reduces the error. However, there is no target output for the hidden nodes so we must use error estimates from the output nodes to derive error estimates for the hidden nodes in order to make adjustments to the incoming weights of the hidden nodes. The term backpropagation stems from the fact that the errors are propagated backwards from the output nodes through the network to adjust the weights in the previous layers. The weights in a BP network are initialized to small nonzero random numbers and then updated after each training sample if an error was found (i.e. the training sample was incorrectly classified). The sigmoid activation function is chosen because the BP training algorithm requires the function to be continuous and differentiable in order for the new weight values to be computed. For a more complete description of the BP training algorithm, see [13].

The function used to make the error estimates defines a surface in weight space. The training algorithm modifies the weights in the network in the direction of the negative gradient of the error surface in an attempt to find a global minimum. There may exist some local minima in the error surface which could cause the network to settle on a set of weights which is not optimum. The use of a momentum term when updating weights can help escape local minima. The momentum is proportional to the previous weight change and tends to keep the weight changes moving in the same direction.

A major drawback of BP networks is the long training times that may be required. Many epochs of the training set may be required for the network to converge on a good solution. There are a number of reasons that BP learning is so slow. One reason is determining the step size by which the weights are changed when the gradient descent is being iteratively performed in weight space. Too small a step size and the network takes too long to converge on a solution, too large a step size and the network may jump over the solution and possibly oscillate instead of converging. Another reason is that all the weights in the BP network are updated at the same time after each training sample, and this may cause all the weights to adjust to reduce the error for one sample and then adjust differently to reduce the error for the next sample, and so on. This type of training may take a while for the network to settle into a good solution for all the training samples. Another problem associated with these networks is determining a good topology. The number of input and output nodes are defined by the problem at hand, but there are no precise rules which state now many hidden layers or hidden nodes in a layer should be selected to achieve good results. There are a few rules of thumb, but for the most part the network topology is determined

by trial and error.

3.4 The cascade correlation neural network

The cascade correlation (CC) ANN attempts to overcome some of the problems associated with standard BP networks. The CC network is self-organizing, so there is no guessing involved in setting up the network topology. The network begins with input and output nodes only, and hidden nodes are added as needed during the training phase. The CC learning algorithm adds hidden nodes to the network one at a time in such a way that the error estimates can be made for each node directly, and therefore error estimates do not have to be propagated backward through the network. The weights are adjusted using the quickprop algorithm[16] which computes for each weight independently the slope of the error surface for the current training cycle and the previous training cycle, and the change that was made in the weight during the last training cycle. The two slopes and the step between them are used to define a parabola which estimates the error surface, and a jump is made to the minimum value of the parabola. The quickprop algorithm reduces the problem of choosing a step size. The CC algorithm allows only a single hidden unit to evolve at a time, and the weights are updated after a single error estimate is made for the entire training set. This strategy lets each hidden unit move directly towards reducing a specific error. The hidden nodes are not all changing at the same time, and this in turn leads to faster learning.

The evolution of a cascade architecture begins with only inputs (including a bias input) and outputs. All inputs are connected to all outputs. The activation function in all output and hidden nodes is a symmetric sigmoid function with an output range of -0.5 to 0.5. When a hidden node is added, it receives input connections from all inputs and all existing hidden nodes. The input weights to the hidden nodes are frozen when the node is added to the network, and only the input weights of the output nodes are updated.

The CC learning algorithm begins with input and output units. The output unit weights are trained with this configuration until learning slows down. The rate of learning is defined by the amount that the error estimate for the network is reduced after each epoch in which the weights are updated. The error estimate is the sum of squared errors for all outputs over the entire training set. When there is no significant error reduction after a number of epochs and there is still a significant error that we wish to reduce, a new hidden node is added to the network, its input weights are frozen, and then all output unit weights are repeatedly trained. This process of training until learning becomes stagnant and then adding a hidden unit is continued until the network converges on a solution, i.e. there is an acceptable error measured for the network.

Hidden units are added to the network according to the following algorithm. The hidden unit receives input connections from all inputs and existing hidden units. The hidden unit outputs are not yet connected to the network. The input weights to the hidden unit are adjusted in an attempt to maximize the sum over all output units of the magnitude of the correlation between the activation value of the unit and the error observed at each output. If the hidden unit correlates positively with the error at a given output, it will develop a negative weight connection to that output. This way the hidden unit is attempting to cancel out some of the error at that output unit. The opposite is true if the hidden unit correlates negatively with the error at an output. After the inputs to the hidden unit are trained, they are frozen and the connections are made to each output unit. The training continues as described above.

The CC ANN demonstrates a number of desirable characteristics. The training times (number of epochs) are shorter compared to BP networks. When running the networks on a serial machine, even greater speedup is observed because (1) error estimates do not have to be propagated backwards through the network, and (2) since the network is built dynamically during training, many epochs are run when the network is smaller than its final size. Another advantage is that the network topology does not have to be determined in advance as with standard BP networks. Since input connections to hidden units are frozen when the unit is added to the network, a CC network can be used for incremental learning in which new information is added to a trained network. For a complete description of the CC network and learning algorithm, see Fahlman's[15] paper.

3.5 The divide and conquer neural network

The divide and conquer network[18] (DCN), like the CC network, is a self-organizing ANN that was developed at USF. The DCN learning algorithm consists of two phases: (1) the divide phase, and (2) the conquer phase, which are done individually for each output during training. Unlike the CC algorithm which creates single node hidden layers, the DCN algorithm will create multiple hidden nodes in the hidden layers if they are needed. Similar to the CC algorithm, the DCN algorithm only trains one unit at a time, allowing the new unit to attempt to correctly classify some of the training samples and eliminating the need to propagate an error signal backwards though the network. Since backpropagation is not required,

a simple delta rule can be used to update the weights in the network, and a threshold activation function is suitable. Training the outputs separately is similar to training different networks for each different class, and DCN allows cells trained for one output to be used while training another output. This is called cell-sharing.

The DCN learning algorithm is now described. The network begins simply with input units corresponding to each input feature, and learning begins in the conquer phase. The conquer phase begins with a new unit being introduced into the network with connections from all inputs and a bias unit. The new unit is trained and if all training samples are correctly classified the algorithm halts, otherwise the divide phase is entered. The divide phase add new hidden units on the same layer as the node introduced in the unsuccessful conquer phase. The divide phase begins by removing all training samples that were correctly classified by the last unit that was introduced to the network. The reduced training set is now augmented by adding neighboring examples. Therefore, for each training sample removed from the training set, a neighbor of that example is created and added to the training set so that the number of training samples remains constant. The neighbor of a sample is defined such that the mean square difference between the original sample and the neighbor is minimal among all other training samples in a different class. The reduced training set is augmented with samples "close" to the samples that are removed for the case that only samples from a single class remain after a divide phase, this way training will be able to continue with samples from more than one class. A set of neighbors is created (one for each training sample) before any training begins, and remains constant throughout training.

When a unit is added during the divide phase, it receives inputs connections from all input units, the bias unit, and all hidden units on previous layers. The weights on these connections are trained on the entire training set with all other connection weights in the network frozen. After a set number of epochs or there is no significant reduction in the error (this is the mean square error of expected values versus actual values for the training set, same as with CC network), the samples that are correctly classified are removed from the training set, the training set is augmented, and a new divide unit is introduced. This divide phase continues until the new unit is able to reduce the training set by at least one example, but has not reduced the overall error by some preselected threshold. When this happens, the conquer phase is entered and a new unit is added at the next level of the network (i.e. a new hidden layer is started). Training is done for the new conquer unit, and it can either correctly classify all training samples from the original training set or the divide phase is entered and processing continues as previously described. When a unit added by the conquer phase is able to correctly classify all the training samples, it becomes an output unit for the network. Figure 4 shows the growth of a DCN network with two outputs.

The DCN algorithm has some desirable features. Like the CC networks, no specification of the network topology is required, as hidden layers and units are added as needed. Avoiding backpropagation of the error signal allows the use of simple learning rules for updating the connection weights. Unlike CC networks, no correlation measure is computed, and the networks created can have multiple hidden units in the hidden layers. This may lead to a higher degree of parallelism in a hardware implementation.

3.6 The traditional k-nearest neighbors algorithm

The K-nearest neighbor (KNN) algorithm is a very simple but powerful method of pattern classification. Unknown patterns are classified based on how similar they are to known patterns. The KNN algorithm computes the distance from an unknown test pattern to every training pattern and selects the K nearest training samples to base the classification on. We use the Euclidean distance in the experiments reported here. The test sample is assigned to the class which has the most samples among the K nearest samples. We are dealing with a two class problem, so the value of K is usually chosen to be odd to ensure that a majority among the two classes can be found.

A variation of the KNN algorithm can be used to bias the decision towards one of the two classes. In this approach, a threshold k less than K/2 can be used for one of the classes instead of a majority vote among the K nearest neighbors of the test sample. This modified KNN rule now states that an unknown test pattern is assigned to a particular class if at least k of the K nearest neighbors is in that particular class. This type of biased decision may be desirable in an application where the penalty for misclassifying one class is much greater than the penalty associated with the misclassification of another class. In our attempt to screen for a sign of cancer, more specifically microcalcifications, we are more concerned with detecting a high percentage of calcifications and are willing to trade off for a lower recognition rate of other objects. By selecting a k for the calcification class, the KNN rule will be more sensitive to calcification detection, and conversely less sensitive to noncalcifications.

4. EXPERIMENTAL METHODS

To obtain the training and test data, a segmentation routine is run on a set of digitized mammograms. The result of the segmentation routine is a template for each image which indicates the locations of possible microcalcifications called candidates. The segmentation routine is designed to locate small, bright spots (a characteristic of microcalcifications) in the raw image. It is important that most individual calcifications, and all clusters of calcifications, be segmented since the overall cluster detection accuracy will rely on the results of the segmentation. Since the segmentation routine will detect objects other than microcalcifications, it will be the job of the classifiers to label the candidates as either yes (a microcalcification) or no. In the case of the probabalistic classifiers, we can provide something analogous to confidence as well. A set of 7 features is systematically chosen and values are computed for each candidate. The feature values are organized into a feature vector, normalized, and written to a data file. Therefore, the training and test data will be 7-dimensional feature vectors which have been normalized between 0 and 1 using the *(value - min)/(max - min)* formula, where value is the feature vector element to be normalized, and max and min are the maximum and minimum possible values for that feature.

4.1 Images

A set of 24 mammograms each containing at least one cluster of microcalcifications were digitized at 70 micron resolution with a DuPont NDT Scan II, Model 35. The images were then divided into a training set of 9 images and a test set of 15 images. The training and test sets were selected so that each would include images with calcification clusters that are embedded in dense parenchymal tissue and therefore more difficult to detect. It should be noted here that due to the small number of images, some bias of our split of the data into training and test sets was unavoidable. An alternate approach would have been to segment all the images and then randomly select candidates from all images for the training and test sets. We feel this would produce higher classification rates, however this could be misleading. "Real world" conditions would prevent using an unknown image which is to be diagnosed as a training image.

4.2 Segmentation routine

A segmentation procedure is used to extract candidate objects from the mammogram images for classification. This routine is able to segment most individual microcalcifications, and all clusters of microcalcifications from the training set images while picking up as few non-microcalcifications as possible. By segmenting candidates from the raw image, the problem becomes one of classification of a hundred or so objects for which good features can be computed, rather than the classification of millions of individual pixels for which a limited number of useful features can be computed.

The segmentation routine used in this initial work is now described. A local contrast image is computed by subtracting from every pixel the average of a 1.13mm (15 by 15 pixels) square region surrounding it. Depending on the maximum pixel intensity in the square region of the local contrast image, the lowest values are discarded, leaving only those pixels with the greatest contrast. If the maximum value in the local contrast image is greater than 15, then a threshold of 10 is selected, otherwise the threshold is 5. Note that this threshold is computed for each pixel in the local contrast image. The result of this is an image with only the locally bright spots remaining. Next, region growing is performed on the local contrast image to group pixels into objects. Pixels that remain at this stage in the local contrast image are assigned a constant value, and the result is a template of candidates which can then be overlaid on the raw image to extract features for each object. In order to reduce the number of candidates segmented, a histogram of the object to background contrast for all objects is created, and only those objects with the highest contrast are retained. This is done by selecting a threshold so that 3% of the total number of segmented objects (with a minimum of 100) are kept. Also, any single pixel objects are eliminated which may correspond to noise spikes. The high resolution digitization process should ensure that microcalcifications are larger than one pixel.

Once the segmentation is done, all objects are manually labelled as either microcalcifications or non-microcalcifications. Thus, we have a 2-class classification problem: (1) microcalcification, or (2) non-microcalcification. The labelling is done to produce a training set, and a test set for which detection accuracy can be computed. An experienced radiologist located the clusters of microcalcifications, and then we were left with the task of labelling the individual microcalcifications.

4.2 Definition of feature space

Once the segmentation has been done, features can be computed for all candidate objects in the template. Since feature selection can be the most crucial component of a pattern recognition system, a systematic evaluation of features that might be used was done to select a subset of features to allow the best possible performance of the classifiers being tested. A literature search on digital mammography, and more specifically the detection of microcalcifications, has resulted in numerous

papers describing features that may or may not be useful. In order to select the features to be used, the set of 24 segmented, labelled images is used. A set of 29 features was chosen to begin with, some from the previously published papers and a few are our variations on previously published features. From the set of 29 features, some were eliminated because a similar feature exists that gives better values. Eighteen features were finally selected for testing. The 18 feature values for all the labelled images are computed, and frequency histograms are plotted for each feature for each class (microcalcifications and non-microcalcifications). If the frequency histograms show decent separation of the two classes, the feature is retained to be included in the feature vector.

Seven features were eventually selected to form the feature vector, they are:
1) Area of object - number of pixels
2) Average grey level of the object
3) Gradient strength of the object's perimeter pixels
4) Root mean square (rms) noise fluctuation in the object
5) RMS noise fluctuation in the 3.5mm by 3.5mm local background surrounding the object
6) Contrast - average grey level of the object minus the average of a two pixel width layer surrounding the object
7) A low order moment based shape descriptor[22]

Only one paper reviewed so far has attempted to select features in a systematic manner. Fox *et al*[4] examined sixty-nine different features and selected the best five using Fisher's linear discriminant. No other paper discusses why certain features were chosen, though some features are obvious. One group did some testing on some shape features[22], but this was to differentiate between malignant and benign calcifications. It seems that a number of groups would determine upper and lower bounds for a feature value associated with microcalcifications, this approach is useful for determining thresholds but they do not indicate how well the feature will separate microcalcifications from other objects that may be detected in earlier stages of the algorithms. This could explain why some attempts at automatic microcalcification detection have only been moderately successful.

4.4 Method of evaluating results

All six methods of classification (LC,QC,KNN,CC,DCN, and BP) require a training set. Results are reported for 5 different sizes of training sets: 100, 200 300 400, and 524. An equal number of training samples from each of the two classes are randomly selected from the set of training images, making sure that we get some samples from every image. For example the training set with 300 samples contains 150 samples of microcalcifications and 150 samples of non-microcalcifications. The training set with 524 patterns (262 from each class) includes all the microcalcifications in the training images. In addition, 5 different sets of random samples are collected for each of the different sizes of training sets, for a total of 25 different training sets. Therefore, there are 5 training sets with 100 samples, 5 sets with 200 samples, etc. For the ANN classifiers, the weights are initialized to random values using 4 different seeds for the random number generator for each of the 25 training sets. There is no correspondence between the weights in the ANNs and any parameter of the KNN classifier. This means each of the ANN classifiers is trained 20 different times for each size of training set, and the KNN classifier is "trained" 5 different times for each size training set. The detection rates for the calcifications and the percentage of false alarms are reported as the mean of all results for the same-sized training sets for each method. When we refer to the detection rate of the noncalcifications, this is simply *100 - false alarm rate*, therefore a high noncalcification detection rate corresponds to a low false alarm rate.

There are some variabilities in each algorithm that one would like to control in order for a fair comparison to be done. The KNN algorithm we described really has two parameters that can be varied, (1) K the number of nearest neighbors to base the decision on, and (2) the threshold k which determines the minimum number of the K-nearest neighbors that must be microcalcifications before an unknown pattern is classified as a microcalcification. We have run the KNN classifier with values of K and k running from 1 to 13 by odd numbers for each training sets for a total of 28 different trials with each of the 25 training sets. For each size training set, an average was obtained for each combination of K and k. The KNN setup which has the maximum cumulative detection rates for the two classes *(%calcification detected + %noncalcification detected)* was selected to compare with the ANN results.

Both dynamic ANNs will add new hidden nodes during training, the BP network requires a fixed topology to be specified

from the beginning. Therefore, the BP network was trained and tested with various numbers of nodes in the hidden layers. The best detection rates were found for 10 hidden nodes in each of the 2 hidden layers, and these are the results reported. The dynamic ANNs are designed to build a topology with enough hidden nodes and layers such that a complete solution (100% correct classification) for the training set can be found. The BP network must search for a solution without changing its configuration, and therefore it may not be possible to get 100% classification rates on the training set. The BP network is trained for 3000 epochs and the error over all training patterns after each epoch is computed, and the network configuration which produces the lowest overall error is saved to be used for the classification results

The LC and QC classifiers are evaluated in three ways. In the first method they are trained on each of the 25 balanced training sets. This training consisted of computing the parameters of the models from the training set, including an appropriate value of T that would produce approximately equal percentages of α (calcifications misclassified) and false alarms (non-calcifications misclassified). Once trained, the true level of α and false alarm are computed using the testing set. In the second method, the models are built using the 25 balanced training sets, but threshold T is set from the testing data. This method better illustrates the relationship between T,α, and false alarm. The third and final method consisted of building the classifiers from the set of all training patterns (all samples from all the training images). As in the first method, the T value is chosen to give equal levels of α and false alarm on the training set, and then the true α and false alarm rate is computed from the testing data. This third method illustrates the ability of the LC and QC to more easily use all available data.

5. RESULTS

The graph in figure 1 compares the calcification detection rates of all six PR methods. The graph in figure 2 compares the percentage of false alarms for all six PR methods. The mean percentages found for each size training set is plotted with error bars for one standard deviation. The BP network produces a large range of detection rates for each size training set, and occasionally could not find a reasonable solution. For the 100 different times the BP network was trained, 19 times the results gave a nearly 100% detection rate for one class, and a nearly 0% detection rate for the other class. These 19 results were discarded when the averages were computed and plotted. If these numbers are included, the average detection rates are 3% to 21% less than reported for different training set sizes. The large error bars on the BP results demonstrate how the BP network may get widely varying results for the same training set with different random starting points for the connection weights or for different training sets of the same size. The BP results seem less dependent on the number of training samples than the other methods. This is evident by the mean values shown in the plots of detection rates. The other three nonparametric methods show the detection rates for both classes more or less increasing with the number of training samples. The LC and QC results are shown for a value of T which produces similar detection rates for both classes as discussed in the previous section.

The DCN network produces the lowest detection rates in most cases for both classes with the average calcification detection rate remaining nearly constant over the different size training sets, and the average noncalcification detection rate increasing steadily as the number of training samples increases. The CC network detection rates continue to increase for both classes as the number of training samples increases. The error bars for the CC network are the smallest for any of the ANN methods, demonstrating fairly consistent results regardless of the random starting points of the connection weights. The KNN method results showed the least variance of any method, and none of the ANNs consistently outperformed it. Worth mentioning is the fact that the best results using the KNN classifier always occurred with the normal majority vote decision rule.

The results reported in figures 1 and 2 for the LC and QC classifiers are for the first method mentioned in the previous section for evaluating the results of these two classifiers. Since these two classifiers allow a desired calcification detection rate to be "dialed in", figures 3 and 4 show power (ROC) curves for methods 2 and 3 (as described in previous section) of evaluating the LC and QC classifiers. These curves show the resulting false alarm rate for a particular desired calcification detection rate. Figure 3 is for when the conditional density functions are modeled using the 25 balanced training sets with the threshold T chosen on the test set. Figure 4 is for when the conditional density funsctions are modeled using the full unbalanced set of training data with threshold T chosen on the training set.

If the sum of the detection rates for the two classes is examined, the best results, averaged over 5 training sets, are for the QC classifier on the 200 sample training sets for which we get 171.1. For the LC classifier on the 200 sample training sets we get an average total detection rate of 169.5. For KNN with K=5 and k=3 for the 524 sample training sets we get 167.4

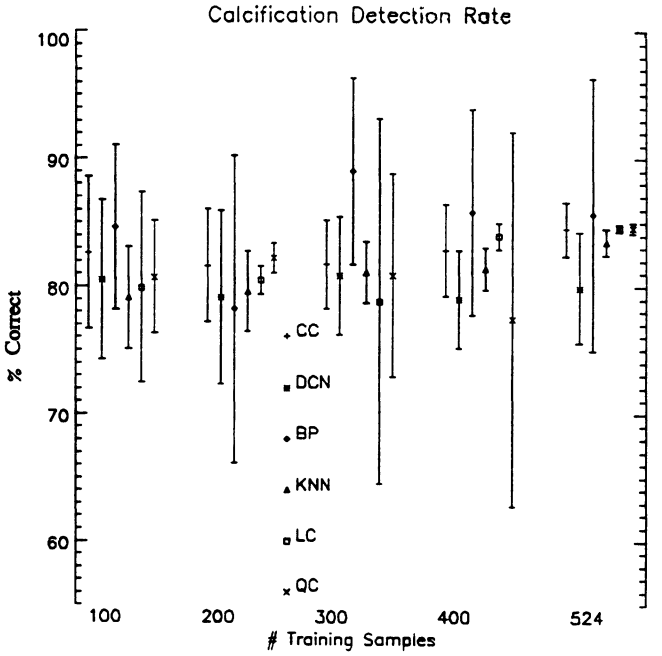

Figure 1 True Positive detection rate of six classifiers for various size training sets.

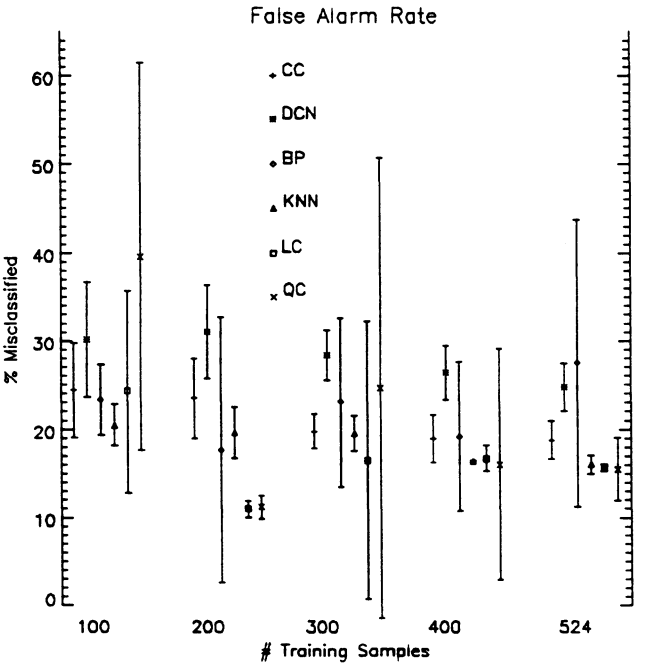

Figure 2. False Alarm rate of six classifiers for various size training sets.

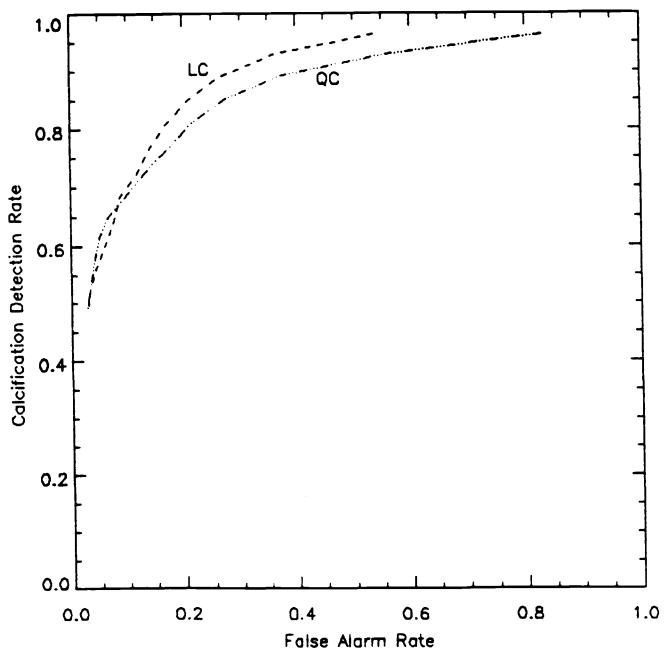

Figure 3. ROC curves for LC and QC classifiers trained on all 25 balanced training sets, and T is set using the test set.

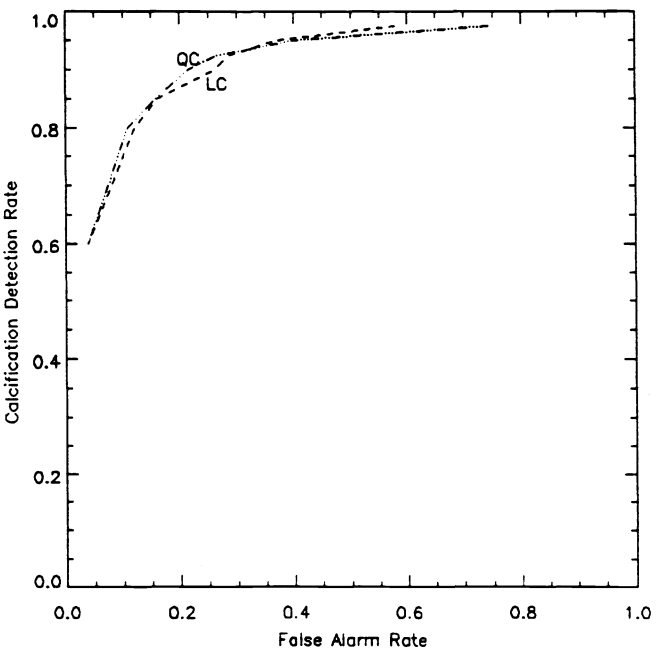

Figure 4. ROC curves for LC and QC classifiers trained on all avauilable training data, and T is set using the training set.

average total detection rate. The BP network averages 166.6 total detection rate for the 400 sample training sets, the CC network averages 165.7 total detection rate for the 524 sample training sets, and the DCN network averages 155.2 total detection rate for the 524 sample training sets.

6. DISCUSSION

The objective of this study is to determine which method or methods of pattern classification is best-suited for automatic microcalcification detection. A complete automated detection system would be composed of several building blocks. It is difficult to determine the best set of blocks all at once. A more practical approach is to work on optimizing each stage of the algorithm separately. From our experiments, we note that among the ANNs the CC and BP networks produce the best overall detection rates on average. The CC network appears to achieve much more consistent results than BP, and many different network configurations may have to be attempted before the best results are found for a BP network. This result suggests continuing experiments with the CC ANN. None of the ANN solutions consistently outperformed the simple KNN classifier, or the two Bayesian classifiers. The LC and QC classifiers not only produced the best results on average, but also have the desirable ability to control the detection rates versus false alarm trade-off. This type of "dial-in" adjustment may prove to be useful for practical use in a clinical situation. It is important to realize that when a classifier is to be used in a clinical setting, many training runs may be required in order to determine the best possible performance that classifier could achieve on a test set. The numbers reported in this paper are for average detection rates over many training runs and do not reflect which methods had isolated training runs which produced the best performance, and would therefore be best for use in a clinical setting.

8. FUTURE WORK

Future work will involve a much larger image database for training and testing. With the exception of the BP network, the PR methods presented here show detection rates increasing as the number of training samples increases. We will continue to increase the number of training samples in future work until maximum detection rates are found. A much larger test set will help determine what can be expected for a larger variation of microcalcification sizes and surrounding tissue densities. A full analysis of the 18 dimensional data including principal component analysis, correlation matrix, and projection pursuit results will be done to produce an optimal set of features and an optimal projection. Some additional features may be included in this analysis including a fractally derived texture information feature. Some more sophisticated methods of statistical PR called kernel classifiers and adaptive mixture classifiers will be examined. A method of segmentation proposed by researchers in our same group using tree-structured nonlinear filters and wavelet decomposition and reconstruction[19] in which the images are first enhanced while preserving details will be compared to the current methods of segmentation to determine its effect on classification accuracy on each of these methods. Full clinical evaluation is now planned using high resolution x-ray film digitizers, and direct digital sensors for stereotactic mammography.

Since locating clusters of microcalcifications is required in an automated screening environment, a simple routine to look for clusters of objects determined by the classifier to be microcalcifications will be implemented. The important numbers that need to be reported here are (1) the true-positive (TP) detection rate of clusters of microcalcifications (how many of the actual clusters did we locate), (2) the false-positive (FP) detection rate of the clusters (how many of the clusters detected were not really clusters), and (3) the false-negative (FN) rate of the clusters (how many actual clusters were not detected at all). The TP and FN rates are inversely related, and these are the most important numbers. It is desirable to catch all abnormalities in a screening. The FP rate can help determine the usefulness of the automatic detection process in a real-world situation. If there are too many FPs and there is not much to be gained from automated screening using these methods.

A screening simulation will be done to evaluate the ability of our methods to detect clusters of microcalcifications in a real mammography screening center. In order to do this, we will take approximately 500 consecutive mammograms from the mammography screening center at H. Lee Moffitt Cancer Center & Research Institute at the University of South Florida. The mammograms will be digitized and run through an automated detection process using one or more of the classifiers described in this paper in the classification step. It is important not to hand pick a set a mammograms to test the screening capabilities of any automatic process designed for this purpose. The mammograms screened should be representative of those that would be encountered in a real world application, and so would produce a truer test of algorithm performance.

8. REFERENCES

1. H.P. Chan, K. Doi, C.J. Vyborny, K.L. Lam, and R.A. Schmidt. "Computer-aided detection of microcalcifications in

mammograms, methodology and preliminary clinical study". Investigative Radiology, 23(9):664-671, Sep 1988.
2.W.G. Wee, M. Moskowitz, N.C. Chang, Y.C. Ting, and S. Pemmeraju. "Evaluation of Mammographic Calcifications Using a Computer Program". Radiology, 116:717-720, Sep. 1975
3. S.H. Fox, U.M. Pujare, W.G. Wee, M. Moskowitz, and R.V.P. Hutter. "A Computer Analysis of Mammographic Microcalcifications: Global Approach". In Proceedings of IEEE Pattern Recognition Conference, pages 624-631, 1980.
4. W. Spiesberger. "Mammogram Inspection by Computer". IEEE Transactions on Biomedical Engineering, BME-26(4):213-219, Apr. 1979.
5. H.P. Chan, K. Doi, S. Galhotra, C.J. Vyborny, H. MacMahon, and P.M. Jokich. "Image Feature Analysis and Computer-aided Diagnosis in Digital Radiography. 1. Automated Detection of Microcalcifications in Mammography". Medical Physics, 14(4):538-548, Jul./Aug. 1987.
6. B.W. Fam, S.L. Olson, P.F. Winter, and F.J. Scholz. "Algorithm for the Detection of Fine Clustered Calcifications on Film Mammograms". Radiology, 169:333-337, 1988.
7. S.L. Olson, B.W. Fam, P.F. Winter, F.J. Scholz, A.K. Lee, and S.E. Gordon. "Breast Calcifications: Analysis of Imaging Properties". Radiology, 169(2):329-332, Nov. 1988.
8. D.H. Davies and D.R. Dance. "Automatic Computer Detection of Clustered Calcifications in Digital Mammograms". Physics in Medicine and Biology, 35(8):1111-1118, 1990.
9. J. Dengler, S. Behrens, and J.F. Desaga. "Segmentation of Microcalcifications in Mammograms". Mustererkennung 1991, Informatik Fachberichte, 290:380-385, 1991.
10. N. Karssemeijer. "A Stochastic Model for Automated Detection of Calcifications in Digital Mammograms". In Information Processing in Medical Imaging, 12th International Conference, IPMI'91 Proceedings, pages 227-238, 1991.
11. I.N. Bankman, W.A. Christens-Barry, I.N. Weinberg, D.W. Kim, R.D. Semmel, and W.R. Brody. "An Algorithm for Early Breast Cancer Detection in Mammograms". In Fifth Anual IEEE Symposium on Computer-based Medical Systems, pages 362-369, 1992.
12. A.K. Jain, "Pattern Recognition". In International Encyclopedia of Robotics Applications and Automation, pages 1052-1063, edited by Dorf, John Wiley and Sons, Inc., 1988.
13. K. Knight, "Connectionist Ideas and Algorithms". Communications of the ACM, 33(11):59-74, Nov. 1990.
14. R.P. Lippmann, "An Introduction to Computing with Neural Nets". IEEE ASSP Magazine, pages 4-22, Apr. 1987.
15. S.E. Fahlman and C. Lebiere, "The Cascade Correlation Learning Architecture". Neural Information Processing Systems 2, pages 524-532, (Ed. D. Touretzky), Morgan-Kaufmann, San Mateo, CA.
16. S.E. Fahlman, "Faster-learning variations on back-propagation: An empirical study". In Proceedings of the 1988 Connectionist Models Summer School, Morgan-Kaufmann Publishers, San Mateo, CA. pages 38-51, 1988.
17. I. Shen, R.M. Rangayan, and J.E. Desautels, "Shape Analysis of Mammographic Calcifications". In Proceedings of the Fifth IEEE Symposium on Computer-Based Medical Systems, June 1992.
18. S.G. Romaniuk and L.O. Hall, "Divide and Conquer Neural Networks", Accepted for publication in the Journal of Neural Networks, to appear in 1993.
19. W.Qian, L.P.Clarke, M. Kallergi, H.D. Li, R.P.Velthuizen, and R.A. Clark, "Tree-structured nonlinear filter and wavelet transform for microcalcification segmentation in mammography". To appear in Proceedings of the SPIE/IS&T Symposium on Electronic Imaging Science and Technology (paper #1905-55), San Jose, CA, Jan31-Feb5, 1993.
20. R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, 1973.
21. E.L. Lehmann, *Testing Statistical Hypothesis*, Wadsworth and Brooks/Cole, 1991.