

Classification of Mass and Normal Breast Tissue: A Convolution Neural Network Classifier with Spatial Domain and Texture Images

Berkman Sahiner*, *Member, IEEE*, Heang-Ping Chan, Nicholas Petrick,
Datong Wei, Mark A. Helvie, Dorit D. Adler, and Mitchell M. Goodsitt

Abstract—We investigated the classification of regions of interest (ROI's) on mammograms as either mass or normal tissue using a convolution neural network (CNN). A CNN is a back-propagation neural network with two-dimensional (2-D) weight kernels that operate on images. A generalized, fast and stable implementation of the CNN was developed. The input images to the CNN were obtained from the ROI's using two techniques. The first technique employed averaging and subsampling. The second technique employed texture feature extraction methods applied to small subregions inside the ROI. Features computed over different subregions were arranged as texture images, which were subsequently used as CNN inputs. The effects of CNN architecture and texture feature parameters on classification accuracy were studied. Receiver operating characteristic (ROC) methodology was used to evaluate the classification accuracy. A data set consisting of 168 ROI's containing biopsy-proven masses and 504 ROI's containing normal breast tissue was extracted from 168 mammograms by radiologists experienced in mammography. This data set was used for training and testing the CNN. With the best combination of CNN architecture and texture feature parameters, the area under the test ROC curve reached 0.87, which corresponded to a true-positive fraction of 90% at a false positive fraction of 31%. Our results demonstrate the feasibility of using a CNN for classification of masses and normal tissue on mammograms.

I. INTRODUCTION

BREAST CANCER IS the most prevalent cancer among women, and it is the leading cause of death from cancer among women of ages 15–54 [1]. Mammography is the most effective method for detection of early breast cancer [2]. However, it is well known that a considerable number of lesions that are visible on mammograms in retrospect are missed by

radiologists. This is due to a variety of reasons including poor mammographic image quality, benign appearance of lesions, and eye fatigue or oversight by radiologists [3]. A computer-aided diagnosis (CAD) scheme can provide a consistent and reproducible second opinion to a radiologist, which may reduce false-negative diagnoses. Furthermore, CAD can also be useful in differentiating malignant lesions from benign ones, thereby decreasing the number of benign cases that are sent for biopsy.

Microcalcifications and masses are the two most important indications of malignancy on mammograms [4]. Previous work on microcalcifications [5] has shown that CAD can significantly improve radiologists' accuracy in detecting clustered microcalcifications. Since masses can be obscured or simulated by normal breast parenchyma [6], detection of masses is a more difficult task both for human observers and computers. In this paper, we report our development of a computerized method for classification of regions of interest (ROI's) as either masses or normal tissue on mammograms using a convolution neural network (CNN).

In the past few years, several researchers investigated the problem of detecting and classifying masses on mammograms. In 1989, Lai *et al.* used image enhancement and template-matching techniques to detect circumscribed masses [7]. They were able to detect all masses on 17 mammograms with an average of 1.7 false positives per mammogram. A limitation of their method was that it applied only to circumscribed masses. In 1990, Brzakovic *et al.* used a fuzzy pyramid linking algorithm to segment and detect ROI's on coarsely digitized (256×256 pixels) mammograms, and developed a classifier to distinguish benign and malignant masses and nontumors based on the morphological features of the segmented objects [8]. More recently, Yin *et al.* used a nonlinear bilateral subtraction technique which identified asymmetries between the right and left breast images, where asymmetries indicated potential masses. Using 46 pairs of mammograms, they correctly identified 95% of the masses with three false positives per image [9]. In 1994, Kegelmeyer *et al.* used local edge characteristics and texture features to detect spiculated masses [10]. Chan *et al.* [11] and Wei *et al.* [12] have developed a classifier using texture features and linear discriminant analysis to classify masses and normal tissue on mammograms, and tested their approach on the same set of ROI's as in this paper.

Manuscript received January 31, 1995; revised November 15, 1995. This work is supported by the USPHS under Grant CA 48129 and the U.S. Army under Grant DAMD 17-93-J-3007 (through subgrant GU RX 4300-803UM from Georgetown University). The content of this publication does not necessarily reflect the position of the Georgetown University or the U.S. Government and no official endorsement of any equipment and product of any companies mentioned in the publication should be inferred. The Associate Editor responsible for coordinating the review of this paper and recommending its publication was A. Burgess. *Asterisk indicates corresponding author.*

*B. Sahiner is with the Department of Radiology, University of Michigan, 1500 E. Med. Center Dr., Ann Arbor, MI 48109-0030 USA (e-mail: sahin@alpha2.rad.med.umich.edu).

H.-P. Chan, N. Petrick, M. A. Helvie, D. D. Adler, and M. M. Goodsitt are with the Department of Radiology, University of Michigan, Ann Arbor, MI 48109-0030 USA.

D. Wei is currently with the Department of Radiology, University of Chicago, Chicago, IL 60637 USA.

Publisher Item Identifier S 0278-0062(96)07196-0.

In recent years, an increasing amount of research efforts has been devoted to the application of neural networks to detection and classification of breast abnormalities. Wu *et al.* used a backpropagation neural network (BPN) to classify ROI's either as normal breast tissue or true microcalcification clusters [13]. Zhang *et al.* employed a spatially invariant neural network for the same purpose [14]. Wu *et al.* also used a BPN to classify benign and malignant tumors based on features perceived by experienced radiologists [15].

The CNN used in this paper is a backpropagation network that operates on images. A CNN was previously used in detection of lung nodules [16] and microcalcifications [17], [18] with some success. In the previous applications, the CNN input was a subsampled version of the ROI itself. In this paper, we generalized the CNN to have multiple input images that can include spatial and texture feature images.

Chan *et al.* and, Wei *et al.* previously investigated the use of image texture features for classification of masses and normal tissue on mammograms [11], [12]. Their results indicated that image texture contains useful information that can be used to effectively distinguish masses from normal tissue. In this paper, we investigated the use of local texture features, which allowed us to characterize texture changes within an ROI more completely. Different images were derived from a single ROI by extracting gray level difference statistics (GLDS) features [19], and spatial gray level dependence (SGLD) features [20], localized on different subregions. Since the features are localized, they vary as the subregion moves over the ROI. By moving the center of the subregion on a rectangular grid over the ROI and computing the features, texture-images were formed. The texture-images were then used as input into a CNN. We investigated the dependence of classification accuracy on the combination of input texture-images and on the CNN architecture.

II. THE CONVOLUTION NEURAL NETWORK

The difference between a CNN and an ordinary BPN applied to image classification is that a CNN operates on images, instead of extracted image features. In this respect, a CNN is similar to the neocognitron of Fukushima [21]. However, unlike the neocognitron, the CNN learning process is very similar to the BPN learning process, i.e., all groups (or planes) in all layers learn at the same time using the backpropagation algorithm.

The basic structure of a CNN is shown in Fig. 1, which depicts a four-layer CNN with two input images, three image groups in the first hidden layer, two groups in the second hidden layer, and one real-valued output. The number of layers and the number of groups in each layer are variable in our implementation. The image propagates from the input to the output by means of convolution with trainable weight kernels.

A. Forward Propagation

Let $H_{l,g}$ denote the g th image group at layer l , and let $N(l)$ be the number of such groups. Image propagation from the input layer ($l = 1$) to the output layer ($l = L$) proceeds as follows. The image $H_{l,g}$ ($l \geq 2$) is obtained by applying a

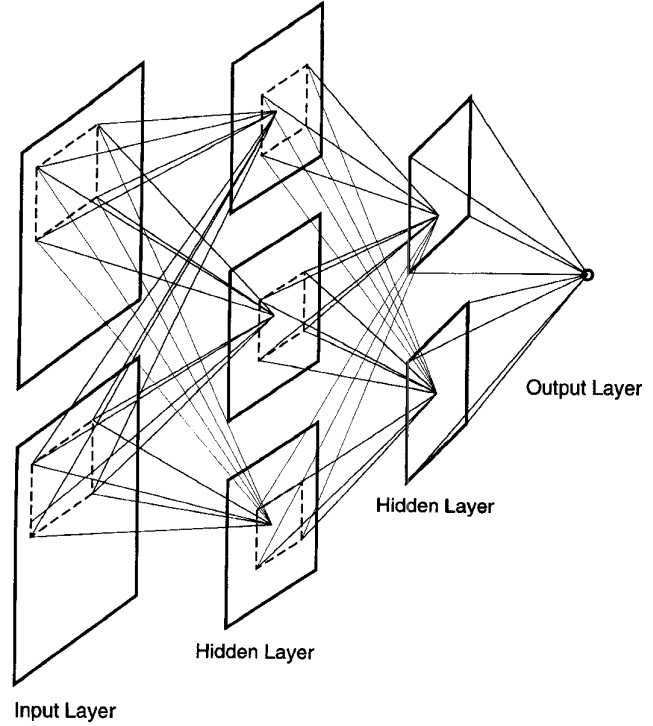


Fig. 1. Basic CNN architecture.

pointwise sigmoid nonlinearity to an intermediate image $I_{l,g}$, i.e.,

$$H_{l,g}(i, j) = \frac{1}{1 + \exp(-I_{l,g}(i, j))}, \quad g = 1, \dots, N(l). \quad (1)$$

The intermediate image $I_{l,g}$ is equal to the sum of the images obtained from the convolution of $H_{l-1,g'}$ at layer $l-1$ with trainable kernel of weights $w_{l-1,g,g'}$. More precisely

$$I_{l,g} = \sum_{g'=1}^{N(l-1)} H_{l-1,g'} * w_{l-1,g,g'} \quad (2)$$

where $*$ denotes two-dimensional (2-D) convolution, and $w_{l-1,g,g'}$ denotes the kernel of weights connecting the g' th group in the $(l-1)$ th layer to the g th group in the l th layer.

The spatial width $S_w(l-1)$ of the weight kernel $w_{l-1,g,g'}$ defines the receptive field for the layer l . The spatial width $S_H(l)$ of an image at layer l is related to the image width at layer $l-1$ by

$$S_H(l) = S_H(l-1) - S_w(l-1) + 1. \quad (3)$$

Consequently, the image width becomes smaller as the layer number increases. The edge effect in convolution is avoided by using this definition. The width of the receptive field of a given node in l th layer is equal to the sum of the kernel widths of the preceding layers minus $(l-2)$. The spatial width of the image at the output layer ($l = L$) is one. The output of the CNN, defined as $O(g) \equiv H_{L,g}(0, 0)$, is thus a real number.

Note that an ordinary BPN is a special case of a CNN. If the weight kernels and the image groups in a CNN are set to be

1×1 arrays, i.e., real numbers, then the weight kernels reduce to ordinary BPN weights and the images reduce to BPN nodes. The equations governing both neural networks are the same.

B. Backpropagation

Similar to BPN, CNN learns through backpropagation. For each training image p (or each training image set p if the input layer has more than one image), one defines a desired-output, $O_d^{(p)}(g)$, where $g = 1, \dots, N(L)$ denotes the output node number. At each training epoch t , training images are applied to the CNN and the actual CNN outputs $O_a^{(p)}(g)[t]$ are computed using (1) and (2). The CNN output error for training image p at training epoch t is defined as

$$E^{(p)}[t] = \frac{1}{2} \sum_{g=1}^{N(L)} (O_d^{(p)}(g) - O_a^{(p)}(g)[t])^2 \quad (4)$$

and the cumulative CNN error during training epoch t is defined as

$$E[t] = \sum_{p=1}^P E^{(p)}[t] \quad (5)$$

where P is the total number of training samples. In a conventional neural network, the weights are updated either after each training sample is presented to the neural network, or after all training samples have been processed (batch processing). The update for a specific weight is proportional to the partial derivative of the neural network output error with respect to that weight, and the constant of proportionality is called the learning rate. It can be shown that both for a BPN and for a CNN, the computation of the partial derivatives can be carried out as a backpropagation process [14].

C. Delta-Bar-Delta Rule

In this paper, to improve the rate of convergence and the stability of the CNN during training, we used batch processing and employed a momentum term and the delta-bar-delta rule proposed by Jacobs [22], as summarized below. The essence of the delta-bar-delta rule is that each weight of a network has its own learning rate, and that these weights are allowed to vary over time. The momentum term is needed to smooth the weight updates over time, thereby resisting erratic weight changes caused by gradient noise [23].

At the end of each training epoch t , the partial derivative $\Delta w_{l,g,g'}(i,j)[t]$ of the cumulative CNN error with respect to each weight is computed using

$$\Delta w_{l,g,g'}(i,j)[t] = \frac{\partial E[t]}{\partial w_{l,g,g'}(i,j)[t]}. \quad (6)$$

After the computation of $\Delta w_{l,g,g'}(i,j)[t]$, the weights are updated using

$$\begin{aligned} w_{l,g,g'}(i,j)[t+1] \\ = w_{l,g,g'}(i,j)[t] - \eta_{l,g,g'}(i,j)[t] \overline{\Delta w_{l,g,g'}(i,j)[t]} \end{aligned} \quad (7)$$

where

$$\begin{aligned} \overline{\Delta w_{l,g,g'}(i,j)[t]} \\ = \theta \overline{\Delta w_{l,g,g'}(i,j)[t-1]} + (1-\theta) \Delta w_{l,g,g'}(i,j)[t] \end{aligned} \quad (8)$$

and $0 \leq \theta < 1$ is called the momentum term.

The individual learning rates are updated using (9) shown at the bottom of the page, where $\kappa > 0$ is called the learning rate increment constant and $0 \leq \phi < 1$ is called the learning rate decrement rate.

Equation (8) computes a moving average with exponential falloff of the partial derivatives over time. If the momentum term is zero, then there is no averaging. If the momentum term is nonzero, then partial derivatives computed during previous training epochs affect the current average, and this helps smooth out the gradient noise over time. In (9), the sign of the moving average is compared to the sign of the latest partial derivative. If the signs are different, learning is oscillatory, and the learning rate for this particular weight is decreased for improved stability. If the signs are the same, stable learning is occurring, and the learning rate can be increased for faster convergence. In [22], it has been shown via several examples that this combination of momentum term and the delta-bar-delta rule considerably accelerates neural network learning.

D. CNN Implementation

It was empirically found that fast and stable convergence was obtained with delta-bar-delta rule parameters set at $\kappa = 0.09$, $\phi = 0.6$, and $\theta = 0.1$. These parameters were fixed at these values through the entire study. Our experience with the CNN indicated that the initial learning rates $\eta_{l,g,g'}(i,j)[0]$ did not have a major effect on CNN learning. These individual learning rates were initialized at

$$\eta_{l,g,g'}(i,j)[0] = 1 \quad \forall l, g, g', i, j. \quad (10)$$

All of the training and test images were first normalized so that the highest pixel value among all images was mapped to 1.0 and the lowest pixel value among all images was mapped to zero. The desired outputs were defined to be 0.9 for mass ROI's and 0.1 for nonmass ROI's.

Initial weights were randomly set to small real numbers using a random number generator. Since overlearning is potentially a problem, CNN performance was monitored over a fixed number of training epochs, and the weights that yielded the best results were chosen as the final weights for a given CNN configuration. Thus, the classification performance of the test set was used as our stopping rule. As discussed in

$$\eta_{l,g,g'}(i,j)[t] = \begin{cases} \eta_{l,g,g'}(i,j)[t] + \kappa & \text{if } \Delta w_{l,g,g'}(i,j)[t] \overline{\Delta w_{l,g,g'}(i,j)[t-1]} > 0 \\ (1-\phi) \eta_{l,g,g'}(i,j)[t] & \text{if } \Delta w_{l,g,g'}(i,j)[t] \overline{\Delta w_{l,g,g'}(i,j)[t-1]} < 0 \end{cases} \quad (9)$$

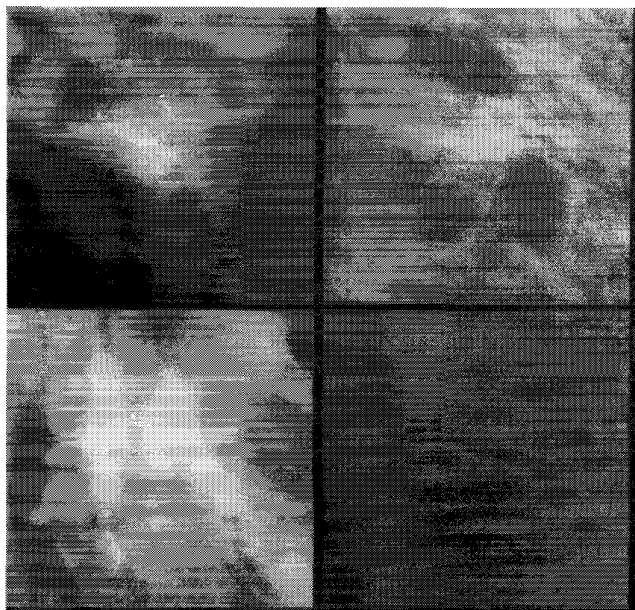


Fig. 2. An example of the mass and normal ROI's selected from one of the mammograms used in this study. The four ROI's are: upper left–mass; upper right–mixed dense/fatty tissue; lower left–dense tissue; lower right–fatty tissue.

Section V, in this study, overlearning did not turn out to be a major problem, i.e., the test results did stagnate, but did not deteriorate as training was continued over thousands of epochs.

Masses on mammograms can be considered to be randomly oriented. Therefore, a CAD scheme is expected to perform similarly on rotations or mirror-images of an ROI. We made use of this fact in our implementation by presenting eight images (or eight image sets if the input layer has more than one image) to the CNN for each ROI in our data set. Four of these images were rotations by 0° , 90° , 180° , and 270° of the original image, and the other four were the rotations of its mirror-image. The CNN output score for an ROI was obtained as the average of the CNN outputs for these eight images. This implementation increased the number of our training ROI's by a factor of eight, and gave us more robust statistics in testing.

III. METHODS

A. Data Set

Our data set in this study consisted of manually extracted ROI's from 168 mammograms which were randomly selected from the files of patients in the Department of Radiology at the University of Michigan who had undergone biopsy. The criterion for inclusion was that the mammogram contained a single biopsy-proven mass. The data set included a mixture of benign ($n = 85$) and malignant ($n = 83$) masses. Of the malignant masses, 45 had spiculated margins. Of the benign masses, six were spiculated. The visibility of the masses ranged from subtle to obvious. The average size (length of the long axis) of the masses, as estimated by the radiologists, was 12.2 mm., and the standard deviation of the mass size was 4.5 mm. The mammograms were randomly divided into training and test groups, each of which contained 84 mammograms.

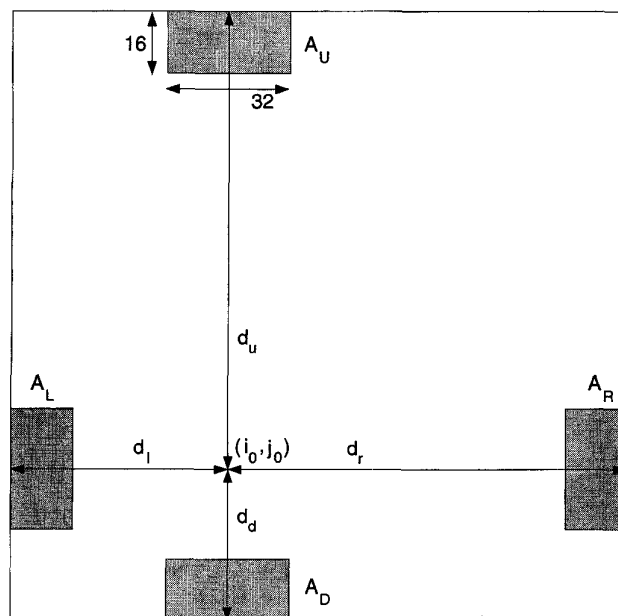


Fig. 3. The averaging boxes and distances used in background correction.

The mammograms were digitized with a LUMISYS DIS-1000 laser scanner at a pixel size of $100 \mu\text{m} \times 100 \mu\text{m}$ and 4096 gray levels. The light transmitted through the film was amplified logarithmically before analog-to-digital conversion. The digitizer had an optical density (OD) range of 0–3.5. It was calibrated so that the OD on film was linearly proportional to the output pixel value in the range of about 0.1–2.8 OD with a slope of 0.001 OD/pixel value. The slope of the calibration curve outside this range decreased gradually.

Four different ROI's, each with 256×256 pixels, were selected from each mammogram by a radiologist experienced in mammography. One of the selected ROI's contained the true mass which was identified by an experienced radiologist and verified by biopsy reports. The remaining three ROI's contained breast parenchyma that was presumed to be normal, with the first region containing dense tissue which could mimic a mass lesion, the second region containing mixed dense/fatty tissue, and the third region containing fatty tissue. An example of each of these ROI's is shown in Fig. 2.

B. Background Correction

The masses superimpose on structured background tissue in the ROI's. In most cases, this background tissue is not uniform over the ROI. For example, one side of the ROI may contain denser tissue than the other side, or, when the mass is close to the outer edge of the breast, one corner of the ROI may contain a nonbreast region. This may reduce the detectability of the mass by a neural network. To reduce this nonuniformity, we developed a background correction method that estimated the background level based on the image intensity in a band of pixels surrounding the ROI.

We estimated a background image $B(i, j)$ from the original ROI as follows. For a given point (i_0, j_0) in the original image, we computed four averages, L, R, U , and D , inside the boxes A_L, A_R, A_U , and A_D shown in Fig. 3. The centers of the

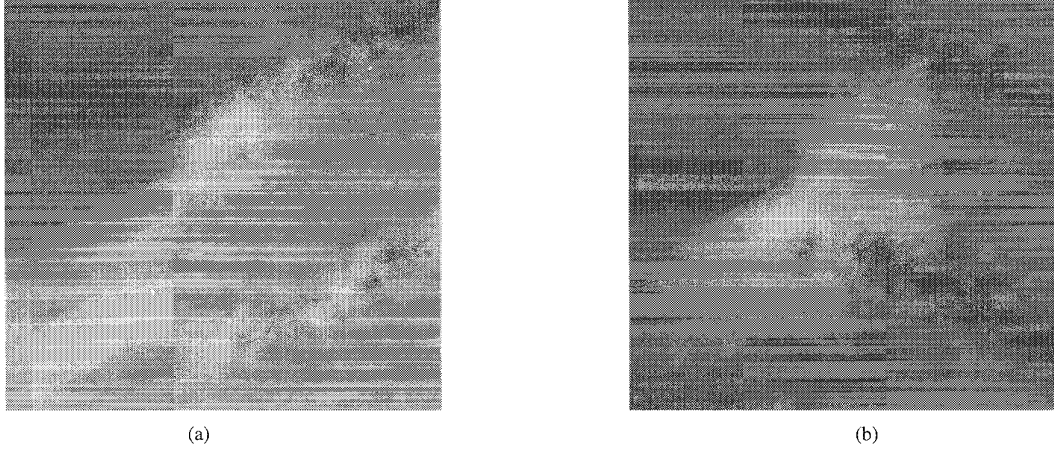


Fig. 4. An example of background correction. (a) Original ROI that contains a malignant mass. (b) Background corrected ROI.

boxes were either at the same row or at the same column as (i_0, j_0) . The box size was 16×32 if it could be placed entirely inside the ROI. Near the corners of the image, the box size was gradually decreased in order to avoid edge effects. For example, if the center of the box was exactly at a corner, then the box size was 16×16 . The background pixel $B(i_0, j_0)$ was interpolated from the averages L, R, U , and D as

$$B(i_0, j_0) = \left[\frac{L}{d_l} + \frac{R}{d_r} + \frac{U}{d_u} + \frac{D}{d_d} \right] / \left[\frac{1}{d_l} + \frac{1}{d_r} + \frac{1}{d_u} + \frac{1}{d_d} \right] \quad (11)$$

where d_l, d_r, d_u , and d_d are the distances between (i_0, j_0) and each side of the image. The background image was then subtracted from the original image, thus reducing the background to near zero.

As mentioned in the Section II-B, the average size of the masses was 12.2 mm, which corresponded to 122 pixels after digitization. Since the masses were placed in the center of the ROI's in the extraction process, very few of the ROI's contained mass tissue in the 16-pixel wide band that was used for background estimation. Out of 168 masses in our database, only four had a long axis longer than 220 pixels, and the long axis in these four cases was not exactly horizontal or vertical. We therefore believe that our estimation essentially excluded information about the mass itself, and included only information about the background. Fig. 4 shows an example of an ROI before and after background correction.

C. Classification with Subsampled Images

The simplest method of classifying mass and nonmass ROI's using a CNN would be to input the background-corrected images directly to the input layer of the CNN. However, the computational cost of inputting 256×256 ROI's into a CNN was prohibitive. We thus had to reduce the image size by averaging adjacent pixels and subsampling. We investigated the effect of reducing the image size to 16×16 and 32×32 . Averaging was performed on nonoverlapping boxes of size 16×16 to obtain 16×16 subsampled images, and of size 8×8 to obtain 32×32 subsampled images.

We investigated the use of a three-layer CNN with a single input image, and a single output node, as shown in Fig. 5. The number of hidden layer groups $N(2)$, and the weight kernel size between the input layer and the hidden layer $S_w(1)$, were variable. For this special case, CNN forward propagation equations simplified considerably. Let $H(i, j) \equiv H_{1,1}(i, j)$ denote the subsampled input image, $W_g(i, j) \equiv w_{1,g,1}(i, j)$ denote the weight kernel between the input layer and the g th group in the hidden layer, and let $W_{g'}(i, j) \equiv w_{2,1,g'}(i, j)$ denote the weight kernel between the g' th group in the hidden layer and the output O . Then, the forward propagation equations (1)–(3) simplified to

$$I_{2,g} = H * W_g, \quad g = 1, \dots, N(2) \quad (12)$$

$$H_{2,g}(i, j) = \frac{1}{1 + \exp(-I_{2,g}(i, j))}, \quad g = 1, \dots, N(2) \quad (13)$$

and

$$O = \frac{1}{1 + \exp \left(- \sum_{g'=1}^{N(2)} H_{2,g'} * W_{g'} \right)}. \quad (14)$$

As mentioned in Section II-D, eight rotated and mirrored images were applied to the CNN consecutively for each averaged-subsampled ROI. The CNN output score for the ROI was obtained as the average of the CNN outputs for these eight images. The CNN output error for training image p was calculated as the square of the difference of the CNN output score and the desired CNN output. Backpropagation with the delta-bar-delta rule was implemented using equations (5)–(8).

After training, the averaged-subsampled images belonging to the test group were applied to the CNN with the trained weights, and the CNN test output scores were obtained using forward propagation. The CNN output scores were used as the decision variable in Receiver Operating Characteristics (ROC) analysis [24] to evaluate the classification performance. ROC analysis evaluates the relationship between the true-positive fraction (TPF) and the false-positive fraction (FPF) as the decision threshold varies. We estimated the ROC curve using the LABROC1 program [25] which assumes binomial

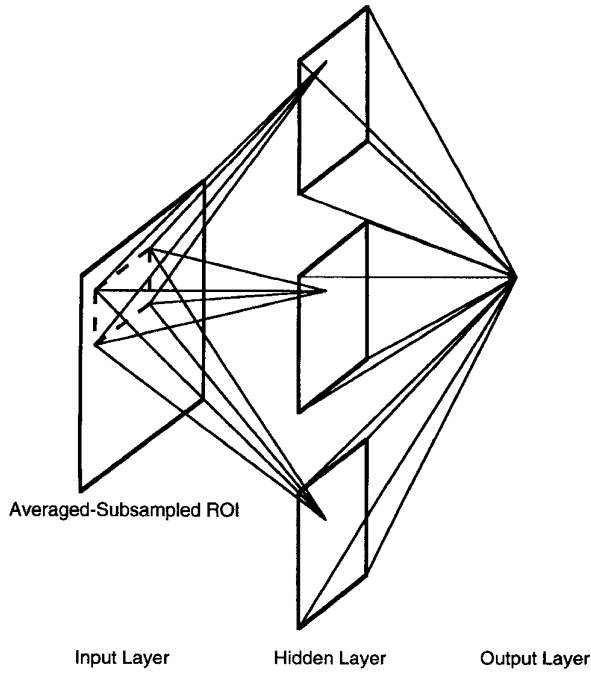


Fig. 5. The CNN architecture used for ROI classification with averaged-subsampled ROI's.

distributions of the decision variable for the normal and abnormal cases and fits the ROC curve based on maximum likelihood estimation. The area under the ROC curve, A_z , was used as an index of classification accuracy. The classification results obtained with the CNN described in this subsection are presented in Section IV-A.

D. Classification with GLDS Texture-Images

1) *GLDS Features*: GLDS features, extracted from the GLDS vector of an image, roughly measure the coarseness of the texture elements in an image. The GLDS vector is the histogram of the absolute value of the difference of pixel pairs separated by a distance d_1 in the horizontal direction and d_2 in the vertical direction. The vector $d = (d_1, d_2)$ is called the displacement vector. As discussed below, the distribution of the elements of the GLDS vector $p_d(k)$ indicate size of the texture element in the image relative to the displacement vector d . GLDS features are extracted by computing some measure of the distribution of the elements of the GLDS vector.

To compute the GLDS vector $p_d(k)$ for a given mammographic ROI $H(i, j)$, and a given displacement vector $d = (d_1, d_2)$, first a difference image is computed as $H_d(i, j) = |H(i, j) - H(i + d_1, j + d_2)|$. The k th entry of the vector p_d is defined as the probability of occurrence of the pixel value k in the difference image $H_d(i, j)$.

If the image texture is coarse, and the length of the displacement vector d is small compared to the texture element size, then the pixels separated by d will usually have similar pixel values. This implies that the elements of GLDS vector will be concentrated around zero, i.e., $p_d(k)$ will be large for small values of k , and small for large values of k . Conversely, if

the length of the vector d is comparable to the texture element size, then the elements of the GLDS vector will be distributed more evenly.

Since the image matrix is discrete, the displacement vector used in feature calculation is usually chosen to have a phase of $\theta = 0^\circ, 45^\circ, 90^\circ$, or 135° . These phases correspond to displacement vectors of $d = (d_0, 0)$, $d = (d_0, d_0)$, $d = (0, d_0)$, and $d = (d_0, -d_0)$, respectively. If image texture is directional, features computed at the same vector magnitude but different phases will convey useful and distinct information. In our case, we did not observe any directional preference in the texture-images that we calculated. Therefore, we averaged textures obtained at the same vector magnitude but different phases. The vector magnitudes at displacement vectors of $d = (d_0, 0)$, $d = (0, d_0)$ and $d = (d_0, d_0)$, $d = (d_0, -d_0)$ differ by a factor of $\sqrt{2}$. For this reason, we averaged the texture features obtained at $\theta = 0^\circ, 90^\circ$ and $\theta = 45^\circ, 135^\circ$ separately. To reduce the number of texture combinations, we used only the averages obtained at $\theta = 45^\circ, 135^\circ$ i.e., we averaged the texture features obtained at $d = (d_0, d_0)$ and $d = (d_0, -d_0)$. In the following discussion, we refer to this average as the feature obtained at a texture distance of d_0 . The effect of different texture distances on classification was evaluated by studying the classification accuracy at texture distances of $d_0 = 2, 4$, and 8 .

In this paper, we used four GLDS texture features, namely, contrast, angular second moment, entropy and mean, which are defined in the Appendix.

2) *GLDS Texture-Images*: Within a selected ROI, there might be several subregions showing different texture statistics, for example, the region inside the mass, the transition region between the mass and the surrounding tissue, and the surrounding tissue. If the texture is computed for the entire ROI, the computation result will be an average of the texture features for the different regions.

One can characterize these feature differences by computing the features in different subregions inside the ROI. In this study, we moved the center of the subregion on a rectangular grid over the ROI, and considered each computed feature as the pixel value of a texture-image at that grid location. The texture-images were then input into a CNN for classification.

Each of the four GLDS features described in the Appendix, namely, contrast, angular second moment, entropy and mean, were used to obtain GLDS texture-images. To obtain a single pixel of a texture-image, one of these features was computed in an $R \times R$ subregion of the ROI. To obtain pixel values of the texture-image at different pixel locations, the center of the subregion was moved over the ROI on a rectangular grid with grid distance G . More precisely, the (i, j) th element of the texture-image was obtained from the $R \times R$ subregion whose upper-left corner was at pixel location (Gi, Gj) in the original image. The computation of a texture-image is illustrated in Fig. 6.

The subregions might or might not overlap depending on the relation between G and R . The size of the texture-image was the smallest integer larger than $(M - R)/G$, where M was the original ROI size. The classification results with GLDS

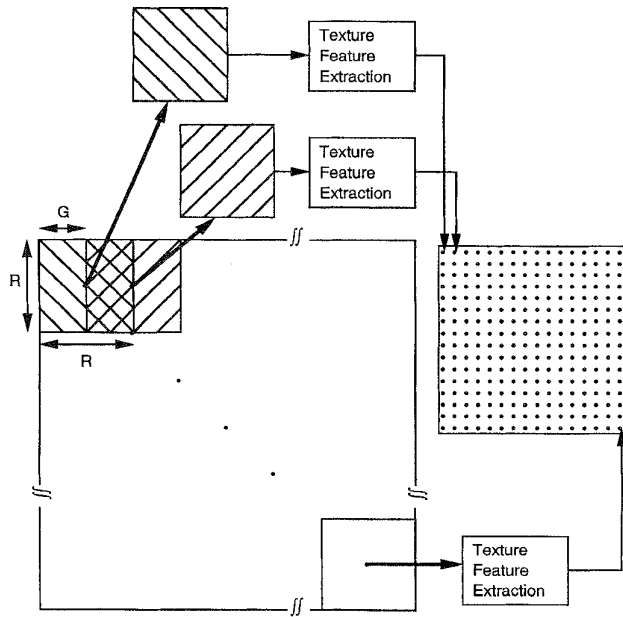


Fig. 6. Computation of texture-images.

texture-images reported in Section IV-B were obtained with $R = 30$ and $G = 15$.

3) *CNN with GLDS Texture-Images*: The CNN architecture employed for classifying mass and nonmass ROI's using GLDS texture-images is shown in Fig. 7. This CNN had a single hidden layer with three image groups, a single output, and two input images. The first input image was a 16×16 averaged-subsampled image that was also used alone for ROI classification with averaged-subsampled images. The second input image was a 16×16 texture-image obtained using one of four GLDS features, contrast, angular second moment, entropy, and mean.

As in the case of CNN with averaged-subsampled images, eight pairs of rotated and mirrored images belonging to each ROI were applied to the CNN consecutively. Since the GLDS features were calculated as the average of texture features at $\theta = 45^\circ$ and 135° , we did not have to recalculate GLDS texture-images for the rotated or mirrored images. We only needed to rotate or mirror the GLDS texture-images, similar to the rotation and mirroring of the averaged-subsampled images. As in the case of CNN with averaged-subsampled images, forward and backpropagation were accomplished using (3)–(8), this time with $N(1) = 2$, $N(2) = 3$, and $N(3) = 1$. Classification accuracy was evaluated using the same methods as in Section III-C.

E. Classification with SGLD Texture-Images

1) *SGLD Features*: A second method of defining statistical texture features is through the SGLD matrix. SGLD features were previously shown to be useful in distinguishing mass ROI's from normal tissue [11], [12]. To compute the SGLD matrix for an image $H(i, j)$, a displacement vector $d = (d_1, d_2)$ is defined. The (k_1, k_2) th element of the SGLD

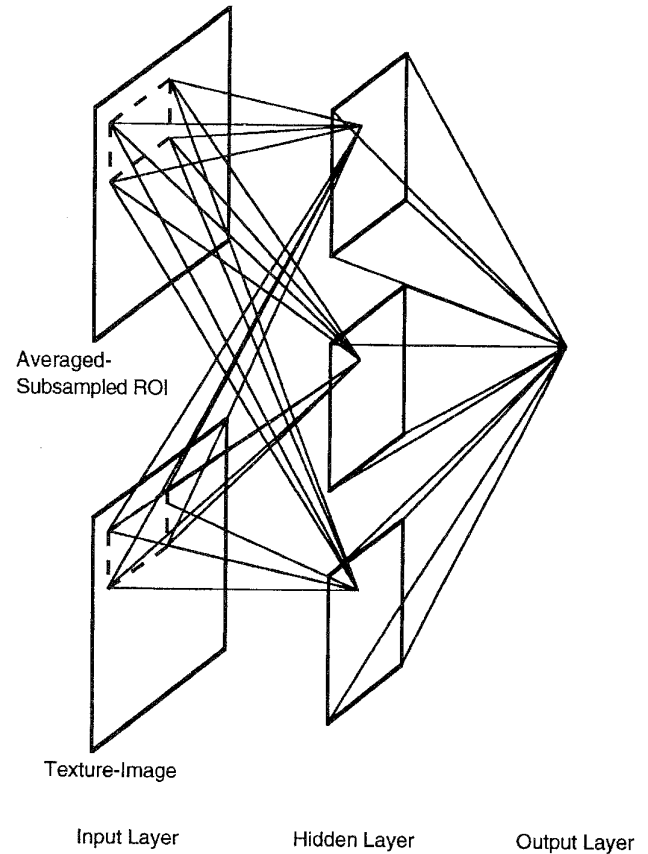


Fig. 7. The CNN architecture used for ROI classification with averaged-subsampled ROI's plus a texture-image.

matrix, P_d , is defined as the joint probability that gray levels k_1 and k_2 occur at a distance of (d_1, d_2) in $H(i, j)$.

SGLD features are affected by the number of bits used to represent the image (bit depth). Images of lower bit depth can be derived from images of higher bit depth by eliminating the least significant bits. The choice of bit depth used in SGLD matrix computation is important because of the trade-off between the gray level resolution and the statistics of the estimated joint probability distribution. If the bit depth is high, then the number of pixels pairs that contribute to an element of the SGLD matrix will be low, and the statistics of the estimated joint probability distribution will be poor. The noise in the least significant bits of the image will also affect the distribution. On the other hand, if the bit depth is low, these two problems are alleviated, but some of the characteristic features of the distribution may be lost due to the reduced gray level resolution. Based on the results of [11], we used a bit depth of 7 bits in SGLD matrix construction.

SGLD features mainly reflect the distribution of the elements in the SGLD matrix. For example, the correlation measure defined in [20] is high when the entries are higher along the main diagonal of the SGLD matrix, and the entropy measure attains its maximum value when all the elements of the SGLD matrix are equal.

TABLE I
CNN CLASSIFICATION PERFORMANCE WITH 16×16 SUBSAMPLED IMAGES

Kernel size	Number of groups	Training A_z	Test A_z
6	4	0.82	0.80
8	4	0.82	0.81
10	4	0.85	0.81
12	4	0.87	0.82
14	4	0.87	0.81
10	3	0.87	0.83
10	6	0.85	0.82
10	8	0.83	0.81

As in the case of GLDS features, we used displacement vectors with phases of $\theta = 45^\circ$ and 135° for SGLD texture feature calculation. These phases corresponded to displacement vectors of $d = (d_0, d_0)$ and $d = (d_0, -d_0)$. Texture features obtained for displacement vectors of $d = (d_0, d_0)$ and $d = (d_0, -d_0)$ were averaged to obtain a GLDS feature at a texture distance of d_0 . The effect of different texture distances on classification was evaluated by studying the classification accuracy at texture distances of $d_0 = 12, 16, 20$, and 24 .

In this paper, we used three SGLD features, namely correlation, entropy, and difference entropy, which were among the best features for the classification of masses and benign tissue in a previous study [11]. The definitions of these features are given in the Appendix.

2) *SGLD Texture-Images*: The computation of SGLD texture-images parallels the computation of GLDS texture-images. Each of the three SGLD features described in the Appendix, namely correlation, entropy, and difference entropy, were used to obtain GLDS texture-images. To obtain a single pixel of a texture-image, one of these features was computed in an $R \times R$ subregion of the ROI. To obtain pixel values of the texture-image at different pixel locations, the center of the subregion was moved over the ROI on a rectangular grid with grid distance G . Fig. 6. describes the computation of texture-images pictorially. The classification results with SGLD texture-images reported in Section IV-C were obtained with $R = 60$ and $G = 13$ for texture distances (d_0) of 12 and 16, and with $R = 75$ and $G = 12$ for texture distances of 20 and 24.

3) *CNN with GLDS Texture-Images*: The CNN architecture employed for classifying mass and nonmass ROI's using SGLD texture-images is the same as that used for classification with GLDS texture-images, and is shown in Fig. 7. This CNN had a single hidden layer with three image groups, a single output, and two input images. The first input image was a 16×16 averaged-subsampled image that was also used alone for ROI classification with averaged-subsampled images. The second input image was a 16×16 texture-image obtained using one of three SGLD features, correlation, entropy, and difference entropy. CNN training and performance evaluation was carried out similarly to Section III-C.

F. Classification with GLDS and SGLD Texture-Images

The CNN architecture employed for classifying mass and nonmass ROI's using both GLDS and SGLD texture-images is shown in Fig. 8. We investigated the use of a three-layer CNN with a three input images, and a single output node.

TABLE II
CNN CLASSIFICATION PERFORMANCE WITH 32×32 SUBSAMPLED IMAGES

Kernel size	Number of groups	Training A_z	Test A_z
11	4	0.81	0.80
16	4	0.84	0.80
20	4	0.84	0.83
23	4	0.84	0.82
20	3	0.84	0.82
20	6	0.84	0.82
20	8	0.84	0.82

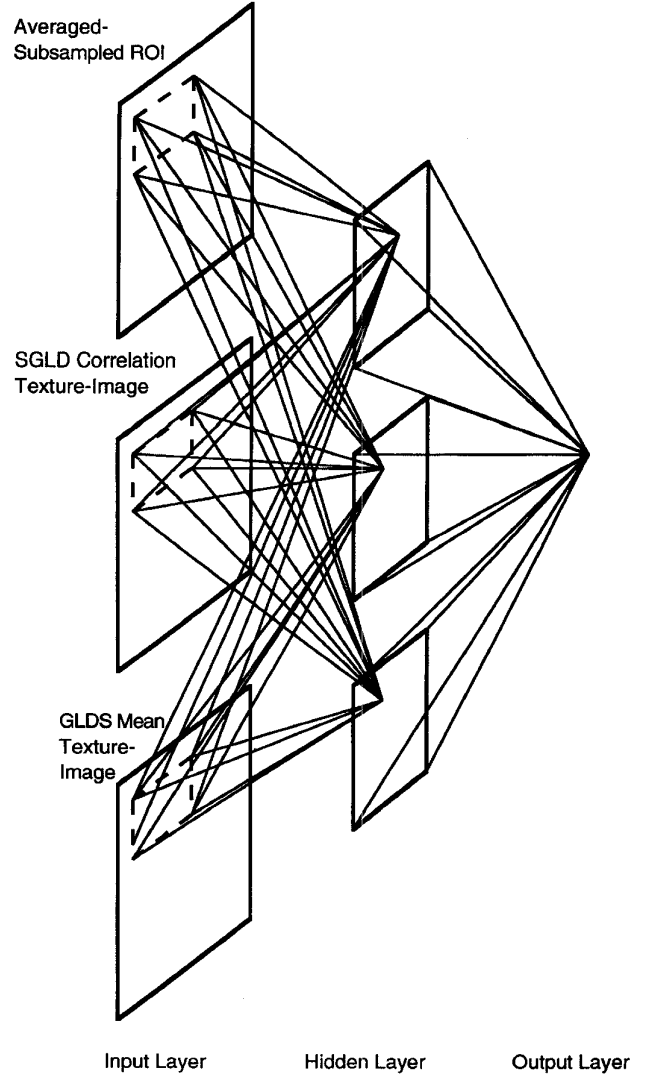


Fig. 8. The CNN architecture used for ROI classification with averaged-subsampled ROI's plus the GLDS mean texture-image and the SGLD correlation texture-image.

The number of hidden layer groups $N(2)$, and the weight kernel size between the input layer and the hidden layer $S_w(1)$, were variable. The first input image was a 16×16 averaged-subsampled image, the second input image was a 16×16 texture-image obtained using the GLDS mean texture-image at a texture distance of $d_0 = 4$, and the third input image was a 16×16 texture-image obtained using the SGLD

TABLE III

CNN CLASSIFICATION PERFORMANCE WITH TWO INPUT IMAGES DERIVED FROM AN ROI. THE FIRST IMAGE IS THE AVERAGED AND SUBSAMPLED IMAGE, THE SECOND IMAGE IS THE GLDS TEXTURE-IMAGE. ASM, CON, G_ENT, AND MEAN STAND FOR ANGULAR SECOND MOMENT, CONTRAST, ENTROPY, AND MEAN, RESPECTIVELY

Feature	$d_0 = 2$		$d_0 = 4$		$d_0 = 8$	
	Training A_z	Test A_z	Training A_z	Test A_z	Training A_z	Test A_z
ASM	0.84	0.82	0.87	0.82	0.86	0.82
CON	0.84	0.82	0.91	0.82	0.86	0.83
G_ENT	0.90	0.84	0.91	0.85	0.89	0.84
MEAN	0.90	0.84	0.90	0.86	0.88	0.85

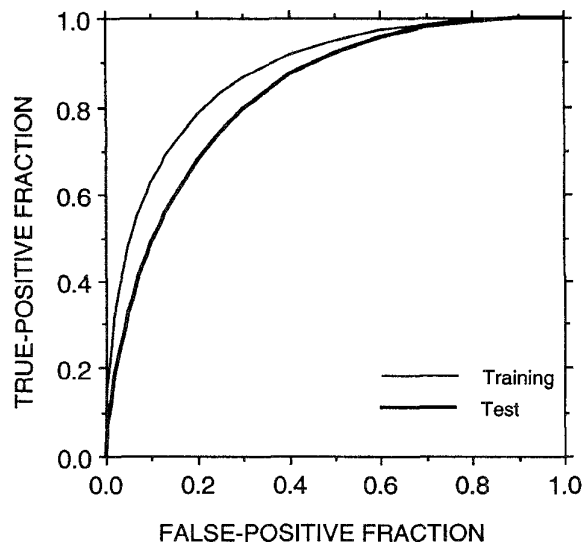


Fig. 9. ROC curve for CNN with the 16×16 averaged-subsampled input image, $N(2) = 3$, and $S_w(1) = 10$. The A_z value was 0.87 for training and 0.83 for test.

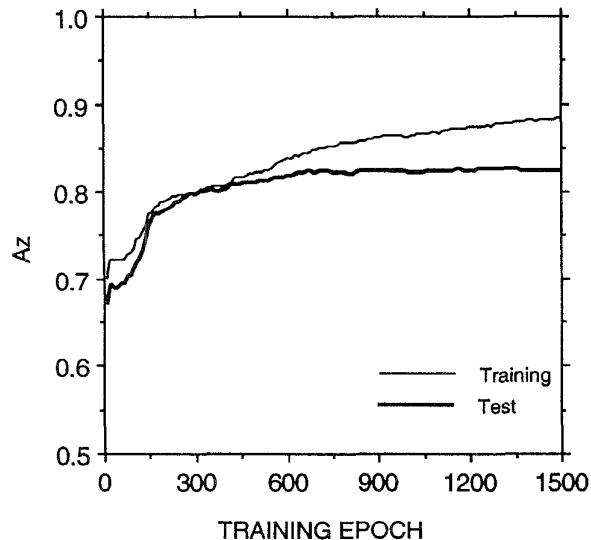


Fig. 10. Training and test A_z values versus training epoch number for the CNN in Fig. 9.

correlation texture-image at a texture distance of $d_0 = 16$. CNN training and performance evaluation was carried out similarly to Section III-C.

IV. RESULTS

A. Results with Subsampled Images

For the purpose of computational efficiency, the image size was reduced by averaging adjacent pixels and subsampling, as described in Section III-C. The resulting 32×32 or 16×16 ROI's were used as inputs to a three-layer CNN with one image group at the input layer, $N(2)$ image groups at the hidden layer, and a single output node. We investigated the effect of varying the number of image groups $N(2)$, and the CNN weight kernel size $S_w(1)$ between the input layer and the hidden layer. The A_z values for the training and test sets are summarized in Table I for 16×16 input images, and in Table II for 32×32 input images, respectively. The training and test ROC curves for the CNN with 16×16 input images, $N(2) = 3$ and $S_w(1) = 10$ are plotted in Fig. 9, and the corresponding learning curves are plotted in Fig. 10.

B. Results with GLDS Features

The results of Section IV-A indicate that the performance was not significantly different i) between 16×16 and 32×32 input images; and ii) among CNN architectures with different

values of $N(2)$ and $S_w(1)$. For this reason, in this subsection, we chose to fix these variables while we studied the effect of the texture feature and distance variables.

All the CNN's in this subsection had two 16×16 input images, $N(2) = 3$, and $S_w(1) = 10$. The first input image was a 16×16 averaged-subsampled image that was also used in the previous subsection. The second input image was a 16×16 texture-image obtained using one of four GLDS features, contrast, angular second moment, entropy and mean. Training and test results are summarized in Table III for texture distances of $d_0 = 2, 4$, and 8.

C. Results with SGLD Features

As in Section IV-B, the CNN's in this subsection had two 16×16 input images, $N(2) = 3$, and $S_w(1) = 10$. The first input image was an averaged-subsampled image, and the second input image was a texture-image obtained using one of three SGLD features, correlation, entropy and difference entropy. We used texture distances of $d_0 = 12, 16, 20$, and 24, because the study in [11] indicated that the best classification accuracy was obtained within this range. Training and test results are summarized in Table IV.

D. Results with GLDS and SGLD Features

In Sections IV-B and IV-C, the CNN architecture was kept fixed as we studied the effect of the texture feature and distance

TABLE IV

CNN CLASSIFICATION PERFORMANCE WITH TWO INPUT IMAGES DERIVED FROM AN ROI. THE FIRST IMAGE IS THE AVERAGED AND SUBSAMPLED IMAGE, THE SECOND IMAGE IS THE SGLD TEXTURE-IMAGE. COR, DIF_ENT, AND S_ENT STAND FOR CORRELATION, DIFFERENCE ENTROPY, AND ENTROPY, RESPECTIVELY

Feature	$d_0 = 12$		$d_0 = 16$		$d_0 = 20$		$d_0 = 24$	
	Training A_z	Test A_z	Training A_z	Test A_z	Training A_z	Test A_z	Training A_z	Test A_z
COR	0.87	0.84	0.84	0.84	0.85	0.83	0.85	0.81
DIF_ENT	0.86	0.82	0.86	0.84	0.86	0.83	0.85	0.82
S_ENT	0.84	0.84	0.86	0.84	0.86	0.83	0.85	0.83

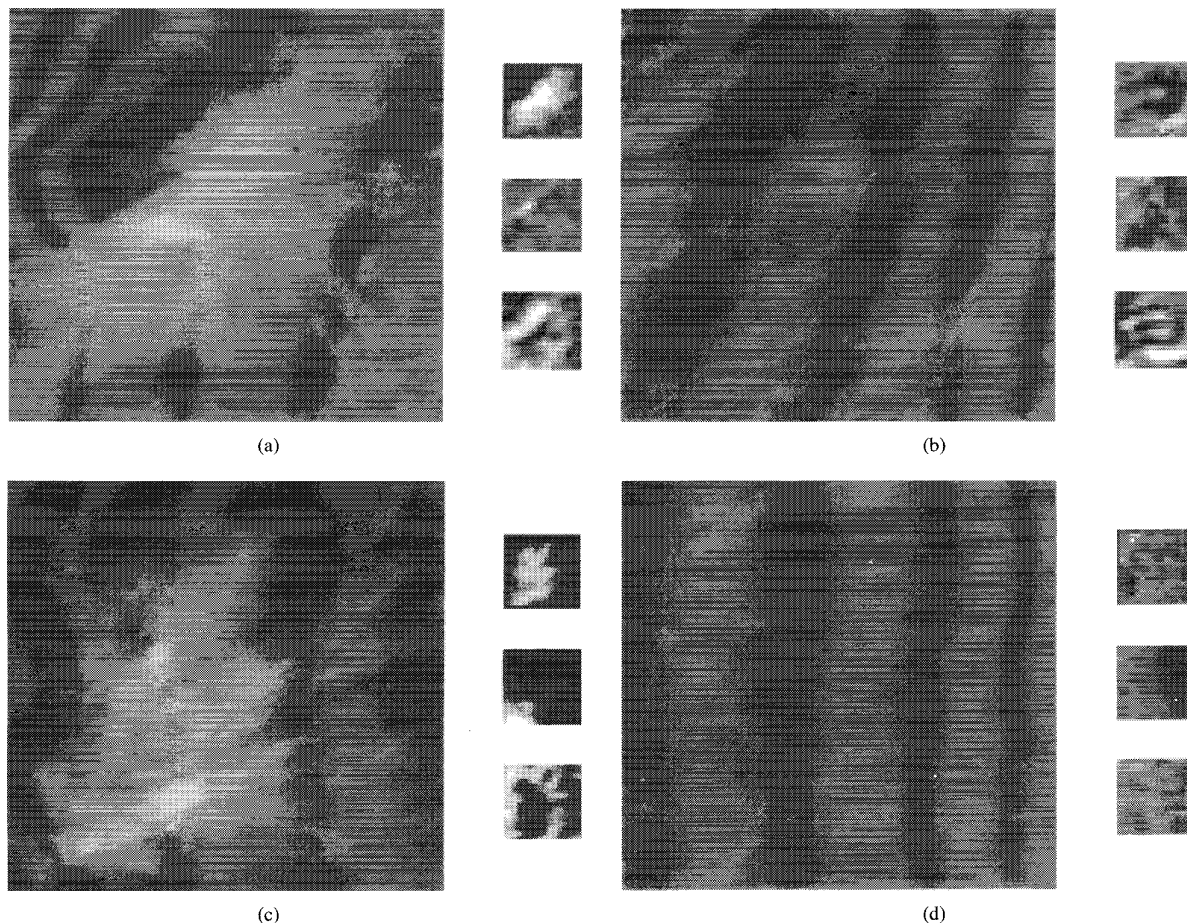


Fig. 11. Background corrected image, subsampled image, GLDS mean texture-image at $d_0 = 4$, and SGLD correlation texture-image at $d_0 = 16$ for (a) a mass ROI, as shown in Fig. 4(b), and (b)-(d) three nonmass ROI's extracted from the same mammogram.

variables for GLDS and SGLD features. In this subsection, we chose one GLDS and one SGLD feature, and studied the effect of the CNN architecture as we did in Section IV-A, but in this case with three input images instead of a single input image. The first input image was a 16×16 averaged-subsampled image that was also used in the previous three subsections. The second image was a GLDS mean texture-image at $d_0 = 4$, and the third was an SGLD correlation texture-image at $d_0 = 16$. These texture-images were chosen because they seemed to yield better classification results than the other texture-images as shown in Tables III and IV. Examples of these three CNN input images, for a mass and three nonmass ROI's extracted from the same mammogram, are shown in Fig. 11, along with the background-corrected ROI's. We investigated the effect of

TABLE V

CNN CLASSIFICATION PERFORMANCE WITH THREE INPUT IMAGES DERIVED FROM AN ROI. THE FIRST IMAGE IS THE AVERAGED AND SUBSAMPLED IMAGE, THE SECOND IMAGE IS THE GLDS MEAN TEXTURE-IMAGE AT $d_0 = 4$, AND THE THIRD IMAGE IS THE SGLD CORRELATION TEXTURE-IMAGE AT $d_0 = 16$

Kernel size	Number of groups	Training A_z	Test A_z
6	3	0.84	0.83
8	3	0.90	0.84
10	3	0.90	0.87
12	3	0.91	0.86
10	4	0.89	0.87
10	6	0.89	0.87
10	8	0.91	0.87

varying $N(2)$, and $S_w(1)$. The A_z values for the training and test sets are summarized in Table V.

The training and test ROC curves for the CNN architecture with $N(2) = 8$ and $S_w(1) = 10$ are plotted in Fig. 12, and the learning curves are plotted in Fig. 13.

V. DISCUSSION

A comparison of Tables I and V reveals that texture-images significantly improve the classification performance. Considering rows with the same number of hidden-layer image groups and the same kernel size in Tables I and V, test A_z values in Table V are 0.04 to 0.06 higher than their counterparts in Table I. The best test A_z value in Table V reaches 0.87, which, as observed from Fig. 12, corresponds to a TPF of 90% at a FPF of 31%. Figs 10 and 13 indicate that as training continued beyond a certain epoch, test A_z fluctuated around a saturation value, and training A_z continued to increase. As the number of CNN input images was increased, we observed a decline in the CNN learning rate, i.e., more training epochs were required for the test A_z curve (bold lines in Figs. 10 and 13) to reach its maximum.

A comparison of different rows in Table I or Table V indicates that the effect of the CNN architecture on classification accuracy is less important than that of the use of texture-images. For example, in Table V, when the kernel size was fixed at ten, and the number of image groups was varied, the test A_z value did not change from its best value of 0.87. Test A_z values within Tables I and V differed by 0.01 to 0.03 when the number of image groups was varied between three and eight, and the kernel size was varied between eight and 12. When we varied the CNN architecture, we did not observe a significant change in the number of training epochs necessary for the test A_z curve to reach its maximum. The overall training time on a computer was longer when the kernel size and the number of image groups were large, since each training epoch took a longer time to run.

One has to study all "reasonable" combinations of CNN architectures and texture feature variables in order to optimize the classification accuracy. However, since CNN training is computationally intensive, this would take an inordinate amount of time. Instead, we attempted to find the "best" combination of features, texture distance, and CNN architecture in two stages, within the constraint of computation time. First, in Sections IV-B and IV-C, we determined which features and texture distances yielded better classification results using a single CNN architecture. Then, in Section IV-D, we varied the CNN architecture while the features and texture distances were fixed. Clearly, this results in a "suboptimal" combination, which, nevertheless, produced satisfactory classification results. It may be possible to improve our results using CNN's that employ more than three input images, more than a single hidden layer, or more than a single output node. It may also be possible to use different techniques to derive different CNN input images from an ROI to further improve the classification accuracy. However, the results of our limited-scale study demonstrate the viability of our approach.

Since a neural network uses an iterative minimization technique in training, the initial state of a CNN is potentially important for training. To obtain an indication about the

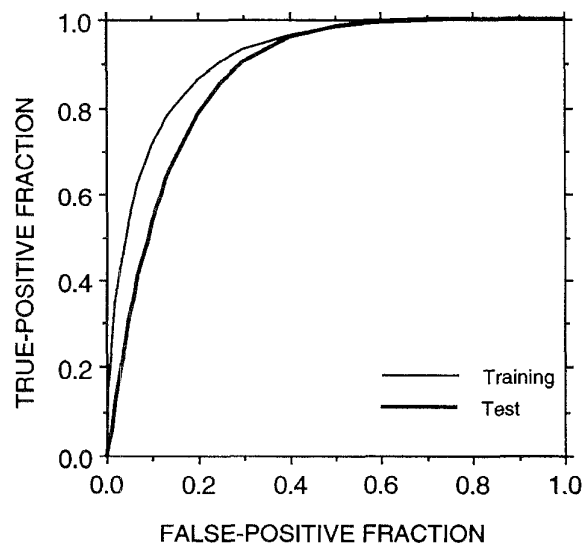


Fig. 12. ROC curve for CNN with three input images, $N(2) = 8$ and $S_w(1) = 10$. The first input image is the 16×16 averaged-subsampled image, the second image is the GLDS mean texture-image at $d_0 = 4$, and the third image is the SGLD correlation texture-image at $d_0 = 16$. The A_z value was 0.91 for training and 0.87 for test.

dependence of CNN performance on initial weight values, we initialized the CNN in the last row of Table V ($N(2) = 8$ and $S_w(1) = 10$) with five different seeds for the random number generator, which produced five different sets of initial weights. After training and testing, we computed the average and standard deviation of the test A_z obtained using these five sets of initial weights. The average A_z was 0.87, i.e. unchanged from the value in Table V, and the standard deviation was 0.002. This indicates that the performance of the CNN that we implemented is consistent in spite of random variations in the initial weights.

Results of Section IV-A indicate that there is no significant difference in classification accuracy between CNN's that operate on 16×16 and 32×32 subsampled ROI's. However, this does not mean that resolution of the ROI does not have any effect on the classification accuracy. It may well be possible that 32×32 subsampled ROI's still do not contain enough detail to improve the classification results. It may also be possible to significantly improve the classification results by applying larger ROI's with better resolution to the CNN. When the computing power becomes available, we will explore the effect of the ROI resolution on classification accuracy.

A shift-invariant neural network (SINN) that is similar to CNN was applied in [14] to detection of microcalcifications on mammograms. CNN and SINN differ mainly in that the test and training outputs of a SINN are images, as opposed to real numbers. The output images of a SINN are processed using thresholding and segmentation techniques before classification is performed. The advantage of SINN is that it yields a spatially invariant output, i.e., ignoring edge effects, if the input ROI is translated, the output is also translated by the same amount. The advantages of CNN in mass detection include i) in training, one does not need to supply a desired image to the CNN that contains the pixel locations of the

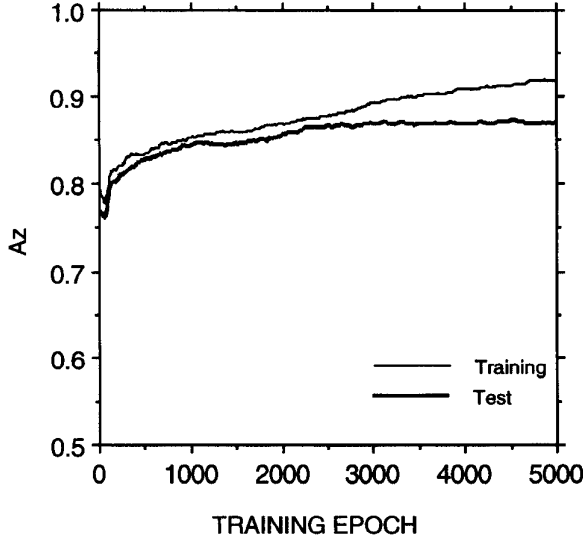


Fig. 13. Training and test A_z values versus training epoch number for the CNN in Fig. 12.

true mass, which may be difficult to obtain in many cases; and ii) after testing, no image processing is required to perform classification: Only a single threshold is used to separate the two classes. In our current application, spatial invariance is not very critical since the masses are centered in the manually extracted ROI's. We are currently developing algorithms which will center the mass in an ROI obtained by an automated detection and extraction program.

In our laboratory, we have previously investigated two other classification techniques using the same ROI set as in this paper [11], [12]. The classification method in [11] employs SGLD texture features obtained at fixed distances, and the method in [12] employs multiresolution texture analysis. The best test results obtained in this paper ($A_z = 0.873$) are better than the best results in [11] ($A_z = 0.823$), and comparable to the best results in [12] ($A_z = 0.859$).

Our interest in CNN as an alternative classifier stems from the fact that different classifiers are potentially better suited to classify different types of masses and normal tissue. It is not yet possible to predict which masses will be more correctly classified by a CNN classifier and which masses will be more correctly classified by multiresolution texture analysis. However, our experiments have shown that combining the outputs of these two classifiers improves classification accuracy. In [26], combining the results of a CNN that operates on averaged-subsampled images (as in Section IV-A), and a classifier that operates on multiresolution texture features, we obtained an A_z value of 0.89 with the same ROI set as in this paper. A more complete analysis of the application of different classifiers to the problem of ROI classification will be published elsewhere.

The long-term objective of this research is to develop a CAD system which will provide a second opinion to the radiologist concerning the presence of lesions on a mammogram. This long-term objective can be divided into two more-easily manageable goals: i) detection of suspicious ROI's on a

mammogram, and ii) elimination of suspicious, but normal ROI's from the ROI's detected in i). This paper deals with this latter goal. When the research on these two goals are integrated, it will be possible to conduct observer studies to evaluate the improvement in radiologists performance when they are assisted by CAD. Although we have not attempted to compare the ROI classification accuracy reported in this paper to that of the radiologists, we suspect that radiologists will perform significantly better than a CNN in classifying the ROI's in this paper. However, the contribution of the CAD system will not be whether the CAD outperforms the radiologists, but rather how much CAD assists radiologists in detecting lesions that would otherwise be missed.

VI. CONCLUSION

We studied the application of a convolution neural network to classification of masses and normal ROI's. CNN input images were derived from the ROI's using i) averaging and subsampling; ii) GLDS feature extraction; iii) SGLD feature extraction. Using a three-layer CNN and three input images derived from each ROI, we obtained an average test A_z of 0.87, which corresponded to an average true-positive fraction of 90% at a false positive fraction of 31%. Our results indicated that the choice of CNN input images is more important than the choice of CNN architecture. Although classification performance needs to be further improved in order for the classifier to be useful in a clinical setting, our study indicates that a CNN can be trained to effectively classify masses and normal breast tissue on mammograms. We are currently investigating the effectiveness of the CNN classifier for differentiation of masses and normal ROI's obtained with an automatic extraction algorithm as a step toward a fully automated computer-aided diagnosis scheme.

APPENDIX

A. GLDS Features

Given a GLDS vector $p_d(k)$ described in Section III-D, the GLDS texture features used in this paper are defined as follows [19], where K is the dimension of $p_d(k)$.

1) Contrast:

$$\text{CON} = \sum_{k=0}^{K-1} k^2 p_d(k).$$

2) Angular Second Moment:

$$\text{ASM} = \sum_{k=0}^{K-1} p_d(k)^2.$$

3) Entropy:

$$\text{G_ENT} = - \sum_{k=0}^{K-1} p_d(k) \log p_d(k).$$

4) Mean:

$$\text{MEAN} = \frac{1}{K} \sum_{k=0}^{K-1} k p_d(k).$$

B. SGLD Features

Given an SGLD matrix $P_d(k_1, k_2)$ described in Section III-E, the SGLD texture features used in this paper are defined as follows [20], where K is the size of $P_d(k_1, k_2)$.

1) Entropy:

$$S_ENT = - \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} P_d(k_1, k_2) \log P_d(k_1, k_2).$$

2) Difference Entropy:

$$DIF_ENT = - \sum_{k=0}^{K-1} P_{x-y}(k) \log P_{x-y}(k)$$

where

$$P_{x-y}(k) = \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} P_d(k_1, k_2), \quad |k_1 - k_2| = k.$$

3) Correlation:

$$COR = \frac{1}{\sigma_x \sigma_y} \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} (k_1 - \mu_x)(k_2 - \mu_y) P_d(k_1, k_2)$$

where

$$\begin{aligned} \mu_x &= \sum_{k_1=0}^{K-1} k_1 \sum_{k_2=0}^{K-1} P_d(k_1, k_2) \\ \sigma_x^2 &= \sum_{k_1=0}^{K-1} (k_1 - \mu_x)^2 \sum_{k_2=0}^{K-1} P_d(k_1, k_2) \\ \mu_y &= \sum_{k_2=0}^{K-1} k_2 \sum_{k_1=0}^{K-1} P_d(k_1, k_2) \\ \sigma_y^2 &= \sum_{k_2=0}^{K-1} (k_2 - \mu_y)^2 \sum_{k_1=0}^{K-1} P_d(k_1, k_2). \end{aligned}$$

ACKNOWLEDGMENT

The authors are grateful to C. E. Metz, Ph.D., for the LABROC1 programs.

REFERENCES

- [1] C. C. Boring, T. S. Squires, T. Tong, and S. Montgomery, "Cancer statistics, 1994," *CA—A Cancer J. for Clinicians*, vol. 44, pp. 7–26, 1994.
- [2] H. C. Zuckerman, "The role of mammography in the diagnosis of breast cancer," in *Breast Cancer, Diagnosis and Treatment*, I. M. Ariel and J. B. Cleary, Eds. New York: McGraw-Hill, 1987, pp. 152–172.
- [3] R. E. Bird, T. W. Wallace, and B. C. Yankaskas, "Analysis of cancers missed at screening mammography," *Radiol.*, vol. 184, pp. 613–617, 1992.
- [4] M. L. Giger, "Computer-aided diagnosis," in *Syllabus: A Categorical Course in Physics Technical Aspects of Breast Imaging*, A. G. Haus and M. J. Yaffe, Eds. Oak Brook, IL: RSNA, 1992, pp. 257–270.
- [5] H.-P. Chan, K. Doi, C. J. Vyborny *et al.*, "Improvement in radiologists' detection of clustered microcalcifications on mammograms: the potential of computer-aided diagnosis," *Investigat. Radiol.*, vol. 25, pp. 1102–1110, 1990.
- [6] L. W. Bassett, D. H. Bunnell, R. H. Jahashahi, R. Gold, R. D. Arndt, and J. Linsman, "Breast cancer detection: One versus two views," *Radiol.*, vol. 165, pp. 95–97, 1987.
- [7] S. M. Lai, X. Li, and W. F. Bischof, "On techniques for detecting circumscribed masses in mammograms," *IEEE Trans. Med. Imag.*, vol. 8, pp. 377–386, 1989.
- [8] D. Brzakovic, X. M. Luo, and P. Brzakovic, "An approach to automated detection of tumors in mammography," *IEEE Trans. Med. Imag.*, vol. 9, pp. 233–241, 1990.
- [9] F.-F. Yin, M. L. Giger, K. Doi, C. E. Metz, C. J. Vyborny, and R. A. Schmidt, "Computerized detection of masses in digital mammograms: Analysis of bilateral subtraction images," *Med. Phys.*, vol. 18, pp. 955–963, 1991.
- [10] W. P. Kegelmeyer, J. M. Pruneda, P. D. Bourland, A. Hillis, M. W. Riggs, and M. L. Nipper, "Computer-aided mammographic screening for spiculated lesions," *Radiol.*, vol. 191, pp. 331–337, 1994.
- [11] H.-P. Chan, D. Wei, M. A. Helvie, B. Sahiner, D. D. Adler, M. M. Goodsitt, and N. Petrick, "Computer-aided classification of mammographic masses and normal tissue: Linear discriminant analysis in texture feature space," *Phys. Med., Biol.*, vol. 40, pp. 857–876, 1995.
- [12] D. Wei, H.-P. Chan, M. A. Helvie, B. Sahiner, N. Petrick, D. D. Adler, and M. M. Goodsitt, "Classification of mass and normal breast tissue on digital mammograms: Multiresolution texture analysis," *Med. Phys.*, vol. 22, pp. 1501–1513, 1995.
- [13] Y. Wu, K. Doi, M. L. Giger, and R. M. Nishikawa, "Computerized detection of clustered microcalcifications in digital mammograms: Applications of artificial neural networks," *Med. Phys.*, vol. 19, pp. 555–560, 1992.
- [14] W. Zhang, K. Doi, M. L. Giger, Y. Wu, and R. M. Nishikawa, "Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant neural network," *Med. Phys.*, vol. 21, pp. 517–524, 1994.
- [15] Y. Wu, M. L. Giger, K. Doi, C. J. Vyborny, R. A. Schmidt, and C. E. Metz, "Artificial neural networks in mammography: Application to decision making in the diagnosis of breast cancer," *Radiol.*, vol. 187, pp. 81–87, 1993.
- [16] S.-C. B. Lo, M. T. Freedman, J. S. Lin, and S. K. Mun, "Automatic lung nodule detection using profile matching and backpropagation neural network techniques," *J. Digital Imag.*, vol. 6, pp. 48–54, 1993.
- [17] S.-C. B. Lo, J.-S. J. Lin, M. T. Freedman, and S. K. Mun, "Computer-assisted diagnosis of lung nodule detection using artificial convolution neural network," in *Proc. SPIE Med. Imag.: Image Processing*, June 1992, vol. 1898, pp. 859–869.
- [18] H.-P. Chan, S.-C. B. Lo, B. Sahiner, K. L. Lam, and M. A. Helvie, "Computer-aided detection of mammographic microcalcifications: Pattern recognition with an artificial neural network," *Med. Phys.*, vol. 22, pp. 1555–1567, 1995.
- [19] J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 269–285, 1976.
- [20] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 610–621, 1973.
- [21] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, pp. 826–834, 1983.
- [22] R. A. Jacobs, "Increased rates of conversion through learning rate adaptation," *Neural Networks*, vol. 1, pp. 295–307, 1988.
- [23] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron, madaline, and backpropagation," *Proc. IEEE*, vol. 78, pp. 1415–1442, 1990.
- [24] C. E. Metz, "ROC methodology in radiographic imaging," *Investigat. Radiol.*, vol. 21, pp. 720–733, 1986.
- [25] C. E. Metz, J. H. Shen, and B. A. Herman, "New methods for estimating a binormal ROC curve from continuously distributed test results," presented at *Annu. Meeting Amer. Statistical Assoc.*, Anaheim, CA, Aug. 1990.
- [26] B. Sahiner, H.-P. Chan, N. Petrick, D. Wei, M. A. Helvie, D. D. Adler, and M. M. Goodsitt, "Image classification using artificial neural networks," in *Proc. SPIE Med. Imag.: Image Processing*, Mar. 1995, vol. 2434, pp. 838–845.