DETECTION OF MASSES ON MAMMOGRAMS USING A CONVOLUTION NEURAL NETWORK

Datong Wei, Berkman Sahiner, Heang-Ping Chan, and Nicholas Petrick

Department of Radiology, The University of Michigan, Ann Arbor

ABSTRACT

A convolution neural network (CNN) was used for classification of masses and normal tissue on mammograms. A generalized CNN was developed that uses multiple images derived from a single region of interest (ROI) as input. CNN input images were obtained from the ROIs using (i) averaging and subsampling; and (ii) texture feature extraction methods on smaller sub-regions inside the ROI. In (ii), features computed over different sub-regions were arranged as texture-images, and subsequently used as inputs to the CNN. Results indicate that using texture-images improves classification accuracy.

1. INTRODUCTION

Computer-aided diagnosis of breast cancer is potentially useful for reducing the number of lesions missed by radiologists at a reasonable cost. Masses on mammograms are good indicators of breast cancer. A number of researchers have attempted to detect and classify them using computer-aided methods [1]. In this study, we investigated classification of regions of interest (ROI) on mammograms either as mass or normal tissue using a convolution neural network (CNN).

A CNN was previously used in detection of lung nodules and microcalcifications with some success [2]. In the previous application, the CNN input was a subsampled version of the ROI itself. In this paper, we generalized the CNN to have multiple image inputs, and investigated using several images derived from a single ROI as the inputs. The images were derived from the ROI by employing various feature extraction techniques [3, 4] localized on different sub-regions.

2. THE CONVOLUTION NEURAL NETWORK

A CNN is a backpropagation neural network. The difference between a CNN and an ordinary backpropagation neural network (BPN) applied to image classification is that a CNN operates on images, as opposed to extracted image features. In this respect, a CNN is similar to the neocognitron of Fukushima [5]. 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.

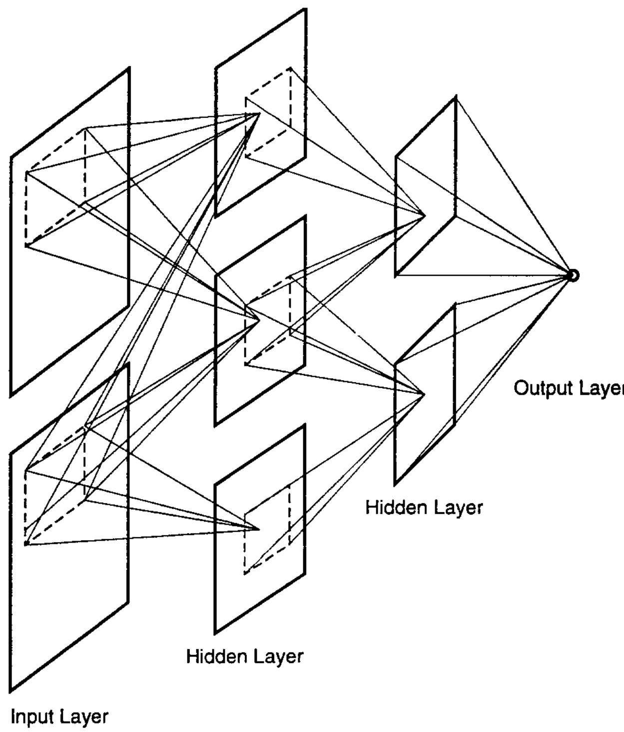

Figure 1: Basic CNN architecture

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 a single real-valued output. The image propagates from the input to the output by means of convolution with trainable weight kernels.

2.1. Forward Propagation

Let $H_{l,g}$ denote the g^{th} image group at layer l, and

0-7803-2431-5/95 $4.00 © 1995 IEEE

let there be $N(l)$ such groups. The 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 pointwise sigmoid nonlinearity to an intermediate image $I_{l,g}$,

$$H_{l,g}(i_1, i_2) = \frac{1}{1 + exp(-I_{l,g}(i_1, i_2))}, \quad g = 1, \ldots, N(l). \tag{1}$$

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 weights $w_{l-1,g,g'}$. More precisely,

$$I_{l,g} = \sum_{g'=0}^{N(l-1)} H_{l-1,g'} ** w_{l-1,g,g'}, \tag{2}$$

where ** denotes 2-dimensional convolution.

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 defined in terms of the image width at layer $l-1$ as

$$S_H(l) = S_H(l-1) - S_w(l-1) + 1. \tag{3}$$

Consequently, the image width decreases as the layer number increases. The width of the receptive field of a given pixel thus increases as the layer number increases. Definition (3) also avoids edge effects in convolution. The spatial width of the image at the output layer ($l = L$) is 1, which means that CNN outputs, defined as $O(g) := H_{L,g}(0,0)$, are real numbers.

2.2. 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 $g = 1, \ldots, N(L)$ denotes the output node number. At each training epoch, training images are applied to the CNN and the actual CNN outputs $O_a^{(p)}(g)$ are computed using (1) and (2). The cumulative CNN output error at training epoch t is defined as

$$E = \frac{1}{2} \sum_{p=1}^{p=P} \sum_{g=1}^{N(L)} (O_d^{(p)}(g) - O_a^{(p)}(g))^2, \tag{4}$$

where P is the total number of training samples. The update for a specific weight is proportional to the partial derivative of the cumulative CNN output error with respect to that weight, and the constant of proportionality is called the learning rate. It can be shown that the computation of the partial derivative can be carried out as a backpropagation process [6].

3. TEXTURE AND TEXTURE-IMAGES

In this paper, we concentrate on texture features extracted from gray level difference statistics (GLDS) vector [3], and from spatial gray level dependence (SGLD) matrix [4].

3.1. GLDS Features

GLDS features roughly measure the coarseness of the texture elements in an image. The GLDS vector $p_d(k)$ for a given image $H(i,j)$ is computed as follows. First, a displacement vector $d = (d_1, d_2)$ is chosen. Then, 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 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 0. 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. One can extract features from the GLDS vector by computing some measure of the distribution of its elements.

Since the image matrix is discrete, the displacement vector used in feature calculation is usually chosen to have a phase of $\theta = 0°$, $45°$, $90°$, or $135°$. In this study, we used the average of features obtained at $\theta = 45°$ and $135°$, *i.e.*, we averaged the texture features obtained at $d = (d_0, d_0)$ and $d = (d_0, -d_0)$. In the following discussions, we refer to this average as the feature obtained at a texture distance of d_0.

In this paper, we used four GLDS texture features, namely contrast (CON), angular second moment (ASM), entropy (G_ENT) and mean (MEAN), whose definitions can be found in [3].

3.2. SGLD Features

SGLD features were previously shown to be useful in distinguishing mass ROIs from normal tissue [7]. 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 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 $H(i,j)$.

In this paper, we used three SGLD features, namely correlation (COR), difference entropy (DIF_ENT) and entropy (S_ENT), whose definitions can be found in [4]. Similar to the calculation of GLDS fetures, we averaged the features obtained at $\theta = 45°$ and $135°$.

3.3. Texture Computation and Texture-Images

Within a selected ROI, there might be several sub-regions showing different texture statistics. For example, part of an ROI may contain an actual mass, whereas the rest of the ROI may contain normal breast tissue. If the texture is computed for the entire ROI, the computation result will be an average of the texture features for a mass and for a nonmass region.

One can characterize these feature differences by computing the features in different sub-regions inside the ROI. In this study, we moved the center of the sub-region 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 thus obtained were then input to a CNN for classification.

4. RESULTS

4.1. The Data Set

Our data set consisted of ROI's of 256 × 256 pixels extracted from screen-film mammograms digitized at a pixel size of $100\mu m$. ROI's were divided equally and randomly into training and test groups. Each group contained 84 mass ROI's which range from obvious to subtle, and 252 normal ROI's which range from fatty to dense glandular tissue.

4.2. Results with subsampled images

Since the computational cost of inputting a 256 × 256 ROI into a CNN is prohibitive, we reduced the image size by averaging adjacent pixels and subsampling. The resulting 32 × 32 or 16 × 16 image matrices were used as inputs to a three-layer CNN. We investigated the effect of varying the CNN weight kernel size $S_w(1)$, and the number of image groups in the hidden layer $N(2)$. The CNN performance criterion that we used was the area under the Receiver Operating Characterictics (ROC) curve, A_z. The results are summarized in Table 1.

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

Table 1.a. CNN classification performance with averaged and subsampled images of 16 × 16 pixels.

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

Table 1.b. CNN classification performance with averaged and subsampled images of 32 × 32 pixels.

4.3. Results with GLDS features

The results of the previous subsection indicate that there is not a significant performance difference (i) between 16 × 16 and 32 × 32 input images; and (ii) among CNN architectures with different values of $N(2)$ and $S_w(1)$. Therefore, we chose to fix these variables while we studied the effect of the texture feature in this subsection.

All the CNNs 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, CON, ASM, G_ENT and MEAN. We varied the texture distance and determined that the best performance is obtained at a texture distance of $d_0 = 4$. Results with this texture distance are summarized in Table 2.

	ASM	CON	G_ENT	MEAN
Training A_z	0.87	0.91	0.91	0.90
Test A_z	0.82	0.82	0.85	0.86

Table 2. CNN classification performance with the averaged and subsampled image and the GLDS texture-image.

4.4. Results with SGLD features

As in the previous subsection, the CNNs 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, COR, DIF_ENT and S_ENT. We varied the texture distance and determined that the best performance is obtained at a texture distance of $d_0 = 16$. Results are summarized in Table 3.

	COR	DIF_ENT	S_ENT
Training A_z	0.84	0.86	0.86
Test A_z	0.84	0.84	0.84

Table 3. CNN classification performance with the averaged and subsampled image and the SGLD texture-image.

4.5. Results with GLDS and SGLD features

In the previous two subsections, the CNN architecture was kept fixed as we studied the effect of the texture feature and distance 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 in Section 4.1, but in this case three input images were used instead of a single input image. The first input image was the 16×16 averaged-subsampled image. The second image was a GLDS MEAN texture-image at $d_0 = 4$, and the third was an SGLD COR texture-image at $d_0 = 16$. We investigated the effect of varying $N(2)$, and $S_w(1)$. Results are summarized in Table 4.

Kernel size	Number of groups	Training A_z	Test A_z
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

Table 4. 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 $d_0 = 4$, the third image is the SGLD COR texture-image at $d_0 = 16$.

5. DISCUSSION AND CONCLUSION

Our results indicate that a CNN can effectively be used for image classification, and that the use of texture-images as CNN input improves classification results. Considering rows with the same $N(2)$ and $S_w(1)$ values in Tables 1 and 4, test A_z values in Table 4 are 0.04 to 0.06 higher than their counterparts in Table 1.

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 very 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 4.3 and 4.4, we determined which features and texture distances yield better classification results using a single CNN architecture. Then, in Section 4.5, 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 by using CNNs that employ more than three input images, more than a single hidden layer, or more than a single output node. Improved CNN results may also be achieved using different types of CNN input images. Our results, although "suboptimal", demonstrate the viability of our approach.

6. REFERENCES

[1] C.J. Vyborny and M.L. Giger, "Computer vision and artificial intelligence in mammography," *American Journal of Roentgenology*, vol. 162, pp. 699-798, 1994.

[2] S.-C. B. Lo, J.-S. Lin, M. Freedman, and S. Mun, "Computer-assisted diagnosis of lung nodule detection using artificial convolution neural network," in *Proceedings of the SPIE*, (Newport Beach, CA), pp. 859–869, June 1992.

[3] J. S. Weszka, C. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, pp. 269–285, 1976.

[4] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, pp. 610–621, 1973.

[5] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, pp. 826–834, 1983.

[6] 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," *Medical Physics*, vol. 21, pp. 517–524, 1994.

[7] 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," Submitted to Medical Physics, 1994.