**BREST CANCER PREDICTION USING DEEP LEARNING:**

To better provide effective cancer detection, a system needs to process 200 to 300 cells per frame, which is impossible through manual tracking [51]. Therefore, the development of effective technologies for breast cancer detection becomes necessary. In contrast, deep learning can be utilized to find patterns in unprocessed data. In recent times, deep learning is a common tool used to detect breast cancer. Deep learning methods have been demonstrated to be capable of diagnosing breast cancer up to 12 months earlier than those using conventional clinical procedures [16]. In addition, the techniques can be used to learn the most pertinent features to best tackle the issue. In recent times, different deep learning-based methods have been introduced for breast cancer diagnosis, which include CNN-, DNN-, RNN-, DBN- and AE-based approaches.

**ABSTRACT:**

The rapid development of deep learning, a family of machine learning techniques, has spurred much interest in its application to medical imaging problems. Here, we develop a deep learning algorithm that can accurately detect breast cancer on screening mammograms using an “end-to-end” training approach that efficiently leverages training datasets with either complete clinical annotation or only the cancer status (label) of the whole image. In this approach, lesion annotations are required only in the initial training stage, and subsequent stages require only image-level labels, eliminating the reliance on rarely available lesion annotations. Our all convolutional network method for classifying screening mammograms attained excellent performance in comparison with previous methods.

**INTRODUCTION:**

The rapid advancement of machine learning and especially deep learning continues to fuel the medical imaging community’s interest in applying these techniques to improve the accuracy of cancer screening. Detection of subclinical breast cancer on screening mammography is challenging as an image classification task because the tumors themselves occupy only a small portion of the image of the entire breast. For example, a full-field digital mammography (FFDM) image is typically 4000 × 3000 pixels while a potentially cancerous region of interest (ROI) can be as small as 100 × 100 pixels. For this reason, many studies13,17,18,19,20,21 have limited their focus to the classification of annotated lesions. Although classifying manually annotated ROIs is an important first step, a fully automated software system must be able to operate on the entire mammogram to provide additional information beyond the known lesions and augment clinical interpretations. If ROI annotations were widely available in mammography databases then established object detection and classification methods such as the region-based convolutional neural network (R-CNN)22 and its variants23,24,25 could be readily applied. However, approaches that require ROI annotations14,26,27,28,29 often cannot be transferred to large mammography databases that lack ROI annotations, which are laborious and costly to assemble. Indeed, few public mammography databases are fully annotated30. Other studies9,10 have attempted to train neural networks using whole mammograms without relying on any annotations. However, it is hard to know if such networks were able to locate the clinically significant lesions and base predictions on the corresponding portions of the mammograms. It is well known that deep learning requires large training datasets to be most effective. Thus, it is essential to leverage both the few fully annotated datasets, as well as larger datasets labeled with only the cancer status of each image to improve the accuracy of breast cancer classification algorithms.

**METHODS:**

Converting a classifier from recognizing patches to whole images To perform classification or segmentation on large complex images, a common strategy involves the use of a classifier in sliding window fashion to recognize local patches on an image to generate a grid of probabilistic outputs. This is followed by another process to summarize the patch classifier’s outputs to give the final classification or segmentation result. Such methods have been used to detect metastatic breast cancer using whole slide images of sentinel lymph node biopsies34 and to segment neuronal membranes in microscopic images35. However, this strategy requires two steps that each needs to be optimized separately. Here, we propose a method to combine the two steps into a single step for training on the whole images (Fig. 1). Assume we have an input patch X∈IRp×q and a patch classifier which is a function f so that f(X)∈IRc , where the function’s output satisfies f(X)i ∈ [0, 1] and Σci=1f(X)i=1 and c is the number of classes of the patches. Here, c = 5 and the classes are: benign calcification, malignant calcification, benign mass, malignant mass and background for each patch from a mammogram. Assume the input patch is extracted from an image M∈IRr×s where p ≪ r and q ≪ s. If the function f represents a convolutional neural network (CNN), then f can be applied to M without changing the network parameters so that f(M)∈IRu×v×c

**NETWORK TRAINING:**

Training a whole image classifier was achieved in two steps. The first step was to train a patch classifier. We compared the networks with pre-trained weights using the ImageNet32 database to those with randomly initialized weights. In a pre-trained network, the bottom layers represent primitive features that tend to be preserved across different tasks, whereas the top layers represent higher-order features that are more related to specific tasks and require further training. Using the same learning rate for all layers may destroy the features that were learned in the bottom layers. To prevent this, a 3-stage training strategy was employed in which the parameter learning is frozen for all but the final layer and progressively unfrozen from the top to the bottom layers, while simultaneously decreasing the learning rate. The 3-stage training strategy on the S10 patch set was as follows:

1.Set learning rate to 10−3 and train the last layer for 3 epochs.

2.Set learning rate to 10−4, unfreeze the top layers and train for 10 epochs, where the top layer number is set to 46 for Resnet50 and 11 for VGG16.

3.Set learning rate to 10−5, unfreeze all layers and train for 37 epochs for a total of 50 epochs.

In the above, an epoch was defined as a sweep through the training set. For the S1 patch dataset, the total number of epochs was increased to 200 because it was much smaller and less redundant than the S10 patch dataset. For randomly initialized networks a constant learning rate of 10−3 was used. Adam42 was used as the optimizer and the batch size was set to be 32. The sample weights were adjusted within each batch to balance the five classes.

**IMAGE CLASSIFICATION:**

Using pre-trained Resnet50 and VGG16 patch classifiers, we tested several different configurations for the top layers of the whole image classifiers. We also evaluated removal of the heatmap and adding two Resnet or VGG blocks on top of the patch classifier layers, followed by a global average pooling layer and the classification output. Model performance was assessed by computing the per-image AUCs on the independent test set.

Resnet-based networks: To evaluate whether the patch classifiers trained on the S1 and S10 datasets are equally useful for whole image classification, the Resnet50 patch classifiers were used. In the original design of the Resnet5039, L ≡ M, N is four times L and K is 3 or more; the L of the current block is also double of the L of the previous block. However, we found this design to exceed our GPU memory limit when it is used for the top layers of the whole image classifier. In the initial experiments, we used instead the same configuration of [512 − 512 − 2048] × 1 for two Resnet blocks on top of the patch classifier. A bootstrapping method with 3000 runs was used to derive 95% confidence intervals for AUCs and AUC differences.