

# Neural Network Analysis of Breast Scan Images

Kay Bean, John Le Houillier, and Denver Massey

*Department of Computer Science, The University of North Carolina at Chapel Hill*

December 7, 2021

## 1 Introduction

Breast cancer is a common cause of death among women worldwide, and early detection is vital in reducing the number of deaths each year [1]. Using an ultrasound imaging machine, doctors are able to take pictures of the interior of the breast tissue in order to check whether the breast is healthy, has a benign tumor or has a malignant tumor. Being able to differentiate between a benign and malignant tumor is vital in saving lives and is currently done manually by doctors. However, would it be possible to use machine learning to aid in identification?

Using a data set of ultrasound images from Kaggle.com [1], we can build a model that would predict whether a breast was healthy or had a benign or malignant tumor. We will be using neural networks as methods of classification. The data set we are using contains 780 labeled ultrasound images of breasts. The images are from women between 25-75 years old and were taken in 2018. The images vary in size, and each is paired with an image showing the boundary of the tumor, a mask. We decided to discard the masks of the tumors since they were not relevant to our project. We wanted a way to classify breast scans without needing to first create image masks.



Figure 1: *Ultrasound of a benign tumor.*

## 2 Methods

We looked at several classification problems with this project. First, we tried classifying each of the types of diagnoses: Normal, Benign, and Malignant. Next, we wanted a model to be able to look at a scan and determine if it was normal or depicted a tumor. Finally, we tried differentiating between benign and malignant tumors. To do all of this, we first prepped the images for analysis. The first step was to remove all masks from our data. Next, due to the images having large size differences, ranging from 190 to 1028 pixels, we cropped all of the images to be 500 by 500 pixels, which was about the average size from the set. Images that were smaller than 500 pixels were given a black border to make up the extra room. After managing the image sizes, we then converted every image to gray scale, because some of the images contained superimposed color. With the size and color of images tweaked, we separated the data for the models and were ready to begin using the images to train our neural networks.

To aid in creating our neural networks, we decided to use the Python Library PyTorch. We converted our prepped images into tensors, since this is standard for PyTorch, and normalized the data set. We then split each of the model's images into two groups, training and testing, where 80% of the images were randomly inserted to the training group, and the remaining 20% were put into the testing group. With this done, we created a basic neural network with three layers and ran each of the data sets through it.

Displeased with our results, we wanted to find a way to improve the accuracy of our models. We lowered the learning rate of the model and added a momentum value of 0.9 to the Stochastic Gradient Descent optimizer. We also generated new samples for the data sets by saving rotated copies of each of the images. We ran the neural network over the Malignant vs. Benign data for 100 epochs and observed the training curve shown in Figure 2.

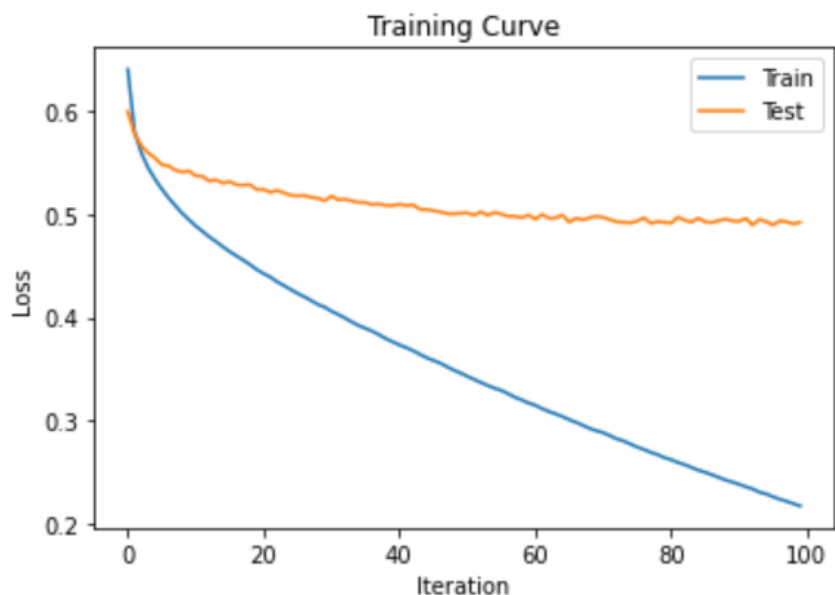


Figure 2: *Training curve for the 3-layer neural network on the Malignant vs. Benign data.*

Using these results as a baseline, we decided to create a convolutional neural network to combat on these problems. The convolutional neural network (CNN) contained eight layers. The first six layers were three sets of convolutional and pooling pairs, followed by two fully connected layers. The parameters for these layers were derived from those of AlexNet, which we studied in class. The convolutional layers break down the images by applying different filters in the hope of being able to identify tumor regions. This helps to discern if the border of a tumor is well defined, indicative of a benign tumor, or not, in the case of a malignant tumor. The final parameters for each convolutional layer (CL) and pooling layer (PL) can be seen in Table 1.

Layer	In Channel	Out Channel	Kernel Size	Stride	Padding
<b>CL 1</b>	1	6	30	5	0
<b>PL 1</b>	6	6	3	2	0
<b>CL 2</b>	6	20	5	1	2
<b>PL 2</b>	20	20	3	2	0
<b>CL 3</b>	20	40	3	1	1
<b>PL 3</b>	40	40	3	2	0

Table 1: *Parameters*

We soon realized that the model was always classifying scans as having a benign tumor. To remedy this, we began using a LeakyReLU activation function with a slope of 0.01 for negative values, rather than the normal ReLU function, in case the gradients were dying out.

### 3 Results

Our results are in the following tables. The first trial compares a neural network (NN) and a convolutional neural network (CNN) determining whether or not there was a tumor in the image given. The accuracy of these networks are given in Table 2.

Table 2: Determining between healthy vs tumor scans.

	NN	NN (Rotated Data)	CNN	CNN (Rotated Data)
Overall Accuracy (%)	85.897	81.731	85.897	81.731
Normal Accuracy (%)	0.00	0.00	0.00	0.00
Tumor Accuracy (%)	100.00	100.00	100.00	100.00

Next, we created networks to distinguish between benign tumor and malignant tumor images. The results of which can be found in Table 3.

Table 3: Determining between benign and malignant tumors.

	NN	NN (Rotated Data)	CNN	CNN (Rotated Data)
Overall Accuracy (%)	73.438	70.930	71.875	62.791
Benign Accuracy (%)	97.826	94.444	100.00	100.00
Malignant Accuracy (%)	11.111	31.250	0.00	0.00

Finally, we created networks to distinguish between a healthy image, benign tumor image, or a malignant tumor image. The results of which can be found in Table 4.

Table 4: Determining between healthy, benign, and malignant scans.

	NN	CNN
Overall Accuracy (%)	68.682	60.620
Normal Accuracy (%)	25.455	0.000
Benign Accuracy (%)	88.366	100.0
Malignant Accuracy (%)	55.172	0.000

## 4 Discussion

When evaluating our results, we need to keep in mind the purpose of our model. The purpose being to aid doctors to diagnose ultrasounds more accurately. Doctors have a false positive rate of around 30% [2] and a false negative rate around 8%[3]. This means that 30% of malignant diagnoses were incorrect and were actually benign while 8% of benign tumors were actually malignant. However, we see that real world doctors tend to report a tumor as malignant even if it is benign, rather than risk a false negative. Looking at our results, many of our neural networks have similar accuracy to that of the real-world doctors. That being said, our goal is to improve diagnosis accuracy.

Various components of our project caused error in our model. One aspect was the data set itself. The images were of different sizes, and, due to this discrepancy, our results were negatively affected. Our solution was to resize the images to a consistent size. This created a risk of cropping out a portion of a tumor for images larger than our chosen size or creating false patterns of black borders for images smaller than our chosen size.

Another issue stemmed from finding appropriate parameters for our models. Parameter selection was limited by our processing power. We were unable to evaluate as many configurations as may have been considered for larger neural network projects, such as AlexNet. Some of our models classified all scans as the same type. This could be problematic if all scans are classified as benign when some are malignant.

There are a few improvements we could make to increase our accuracy. Starting with addressing the variation in image sizes. More research could be done to find machine learning techniques that can evaluate differing images sizes. This would eliminate a need for cropping. In contrast, additional machine learning techniques could be used to locate the tumor, and the image could be cropped without deleting the tumor. Another option would be to create a program that masks the images similar to the provided masked images in the data set. To improve this project, we could implement a different neural network to detect where the tumor is, and create a mask. This would completely solve our issues with the data set, as long as our detection of the tumor location was accurate.

## References

- [1] Al-Dhabyani W, Gomaa M, Khaled H, Fahmy A. Dataset of breast ultrasound images. data in brief. <https://www.kaggle.com/aryashah2k/breast-ultrasound-images-dataset>, Feb 28, 2020.
- [2] B. Archie, H. Gilbert Welch. *Effect of Three Decades of Screening Mammography on Breast-Cancer Incidence*. Nov 22, 2012.
- [3] *SENSITIVITY, SPECIFICITY, AND FALSE NEGATIVE RATE FOR 1,682,504 SCREENING MAMMOGRAPHY EXAMINATIONS FROM 2007 - 2013*. March 23, 2017.