Mammogram Classification Using Convolutional Neural Networks

Henry Zhou
Henry.Zhou@tufts.edu

Yuki Zaninovich
Yuki.Zaninovich@tufts.edu

Chris Gregory
Chris.Gregory@tufts.edu

Abstract

Fine needle aspiration biopsies are a common method of testing for cancerous cells. As a surgical procedure, FNA biospies can be both invasive and costly. We explored the viability of analyzing mammograms (X-ray images of breasts) using an image recognition machine learning algorithm to assist in classification of benign and cancerous abnormalities as well as abnormality detection. Specifically, we analyzed the effectiveness of convolutional neural networks (CNN) in determining the existence of breast abnormalities in mammograms. If there is an abnormality, it also classifies it as either benign or malignant. In an effort to maximize our results working with a relatively small dataset, we employed a set of pre-processing techniques to strengthen our classifier, including image transformation and classification redistribution. We tested a variety of architectures to find the one that produced the best recall figures, which we deemed to be the most important measures of performance within the context of malignancy detection.

1 Introduction

Breast cancer is one of the leading causes of death for women globally. With over 1.7 million new cases diagnosed in 2012, breast cancer is the most common form of cancer worldwide. In 2016, there will be an estimated 246,660 new cases of invasive breast cancer, 61,000 cases of non-invasive breast cancer, and 40,450 breast cancer deaths [10]. In this paper, we propose using an image recognition system that utilizes a convolutional neural network in order to detect and classify abnormalities in mammograms. In general, a mammogram is classified as either normal, benign (non-cancerous abnormality), or malignant (cancerous abnormality). In practice, it can be difficult for patients to obtain a quick diagnosis regarding a breast abnormality solely from a doctor or radiologist's examination of a mammogram. A breast biopsy is usually required before a medical professional can make a diagnosis. As a surgical procedure, there exist risks and side effects such as chronic pain and infection that may result from receiving a biopsy. Moreover, scheduling a biopsy, finding a doctor, and waiting for the lab results all prolong the time required to make a diagnosis, which may give the cancer enough time to leave irreversible effects on the patient's health.

For our project, we analyzed the effectiveness of convolutional neural networks in both detecting abnormalities in mammograms and classifying them as benign or malignant. To do so, we utilized TensorFlow, Googles open source Machine Learning library. With TensorFlow's CNN module we trained a classifier on mammogram data sourced from the mini-MIAS database [11]. In order to optimize our classifier, we explored various methods of training by benchmarking classifiers with different factors such as number of hidden layers, kernel size, learning rate, etc. Consequently, we iterated upon our learning model through a combination of statistical and machine learning theory, domain knowledge of our data, and trial and error. We discuss theory behind convolutional neural networks as well as previous research related to CNNs in Section 2. Section 3 explains our methodology for both preprocessing our data and selecting the architecture for our CNN. Section 4 presents the results we observed when using our classifier, and Sections 5 and 6 cover some general discussion regarding the project as well pathways of future study.

2 Background and Previous Work

Some previously completed research has aided us in directing our methods and augmenting our understanding of CNN image classification. Krizhevsky et al's paper ImageNet Classification with Deep Convolutional Networks is a highly cited paper in the field of image classification and deep neural networks [4]. The researchers demonstrated low error rates on top 1 and 5 results on the ImageNet LSVRC-2010 test set using a CNN with various overfitting reduction, training, and performance boosting techniques. The paper was the main inspiration behind our work and served as supporting evidence that CNNs are an effective method for image classification. Many of the techniques used by Krizhevsky et al have been implemented in our neural network. For example, the preprocessing techniques we use to transform our dataset were inspired by their work. Like Krizhevsky et al, we used subsampling and rotation to reduce overfitting of our classifier for greater generalization. We also used non-saturating nonlinearities as our activation function for each neuron (ReLUs) as presented in the paper in order to hasten training, an idea that was also presented in Jarrett et al [1].

Sahiner et al introduced us to some important concepts in image preprocessing for machine learning [9]. Namely, training a convolutional neural network on regions of interest rather than full image instances helps guarantee that the learned features are those of abnormalities, not some common feature of normal mammograms. This subsampling technique used by Sahiner et al also led us to realize the importance of reducing the image dimensions, and by extension the network layer size, to boost computational performance. Techniques like simple cropping and averaging adjacent pixels, allowed for this transformation. In turn, better performance helped us iterate over our CNN architecture faster and optimize our network parameters. In addition, this paper introduced us to the idea that rotation and mirroring of mammograms can help expand our constrained dataset, reduce overfitting, and increase how generalized our neural network is.

Another paper, produced by Lo et al, helped us understand that the number of network layers used should depend on the complexity of the pattern to be evaluated and that two or three network layers are usually adequate for the simplicity of the patterns indicative of mammogram abnormalities [5]. This paper also helped us understand that probabilistic output with a CNN is possible via one-hot encoding with multiple neural network output neurons. Lo et al chose to let a few of those output neurons indicate the region where the probabalistic outcome is inconclusive. While Lo et al used 2×2 average pooling to extract features, we opted to preserve the 2×2 structure, but to use maximum pooling in the hopes of extracting the most important features at each pooling stage. Both the paper by Lo et al and Sahiner et al recommended using an ROC (receiver operating characteristic) curve to analyze the accuracy of results, especially given that the distribution of class labels in our dataset (as in many medical datasets) is skewed more towards normal samples rather than malignant.

All of these considerations were important in the evolution of our classifier and analysis of the results. Not every suggestion was applied to our final model, but many of the concepts introduced in these papers led us to make decisions that produced better computational performance and favorable results.

3 Methodology

3.1 Convolutional Neural Network

Convolutional neural networks are effective for image classification problems because the convolution operation produces information on spatially correlated features of the image. Convolution is performed by initializing a square matrix with specific values. This matrix, or kernel, is then applied to each pixel in an image. For each pixel in an image, the kernel multiplies the pixel and its adjacent pixels that the kernel covers by their corresponding kernel values. The products are then summed and this value is set as the pixel value in the convolved image at the initial pixel's location.

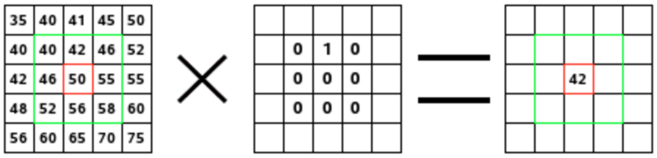

Figure 1: Convolution of a matrix with a 3 x 3 kernel [15]

As a result of convolving, the image is filtered for specific features and those patterns are enhanced to produce a new overall effect. For example, convolving may result in edges becoming more prominent or the entire image becoming more blurred. This can be valuable in extracting specific features unique to certain images that indicate a particular class. After convolution, an image's specific identifying features may be more readily learned by a fully connected neural network than they would be without the convolutional step. Our CNN takes an image and convolves various types of kernels over the image, creating different output nodes that later get fed into more convolutional or fully connected layers. More informative kernels that help with classification become the more active nodes.

Though one convolutional layer can only detect elementary features, due to the nature of convolution, feeding the output of one convolutional layer to another allows for high-order feature extraction. For example, an initial kernel may be optimized to extract edges within the initial convolutional layer, and a second kernel may soften more organic shapes in the next layer, and so on.

Our CNN also uses a technique called max pooling. Max pooling takes the output of the convolution and splits it up into tiles. We chose our tiles to be 2×2 pixels each. Only the largest value from each tile is used in the next layer of the network. In the past many researchers using CNNs used average (or mean) pooling. This can be seen in the work of Lo et al. The reasoning behind average pooling makes sense in that taking an average of the pixels will assure that no information is completely lost in the pooling step. However, increasingly often max pooling is used to extract the most prominent groups of features from each convolution. This means that the output into later layers is filtered for the most informative patterns relating to the problem domain. Another consequence of pooling is that the input is reduced in size, which reduces computation time by reducing the number of inputs to the fully connected neural network.

As with most CNNs, ours uses back propagation in or-

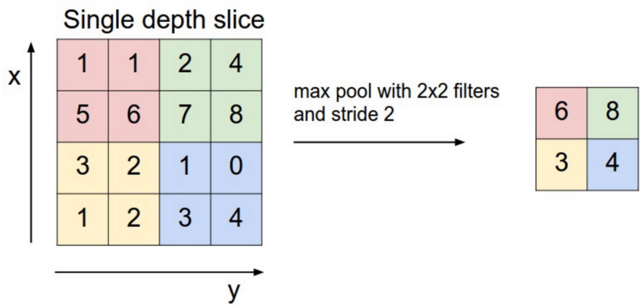

Figure 2: Visualization of Max Pooling [16]

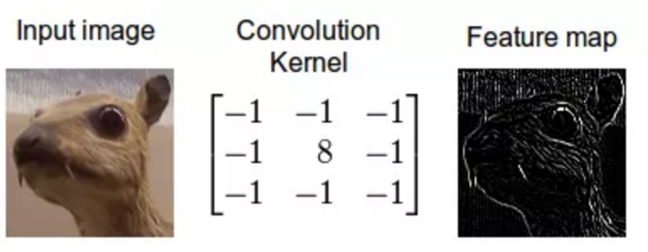
Input image
Convolution Kernel
$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
Feature map

Figure 3: Example of edge extraction effect on an image due to convolution [14]

der to update weights to be closer to their optimal values. This means that every time input is passed through the weighted layers of the network, an error value is calculated for the expected output. That error is propagated back through the network to update the weights that contribute most to the error. After multiple iterations or steps, the weights learn by being updated to make more and more accurate predictions based on the training data expectations.

There are several parameters of CNNs that can help optimize training time, such as learning rate and stride. Learning rate determines the rate at which neuron weights are updated during backpropagation. A low learning rate will likely yield a higher accuracy (though it could get stuck in local optima) as it will be able to optimize weights to a higher significant figure, but the convergence time will be non-trivial. Conversely, a learning rate that is too high may not get close enough to the global optima and diverge. Stride dictates how many pixels of the input image the kernel slides over and skips between individual convolutions. This effectively reduces both the number of convolutions in the current layer and the dimensions of the image outputted by the convolution, meaning the reduction in convolutions and processing time in future layers will be of an even larger factor.

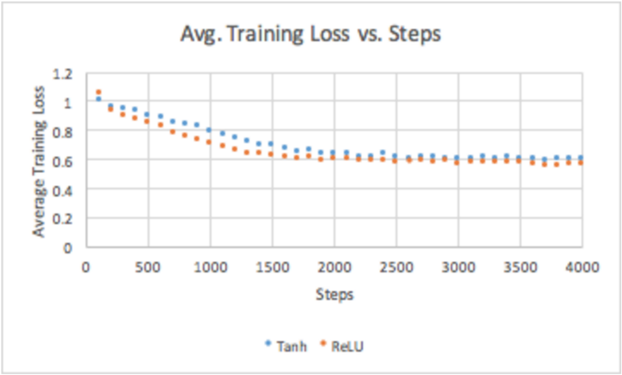

Figure 4: Tanh vs ReLU average training loss over steps

3.2 ReLU Activation

In a neural network, neuron output or activation function generally has the form $f(x) = tanh(x)$ or the sigmoid function for some input x. These neuron activation functions are by definition saturating because they ultimately map their input to an output between a fixed range such as $[-1, 1]$. Neurons that use non-saturating activation functions such as $f(x) = max(0, x)$ are called Rectified Linear Units and can be used in a Convolutional Neural Network. Theoretically, CNNs that train on ReLUs versus neurons using $f(x) = tanh(x)$ train faster, ceteris paribus. Given that training a CNN is a computationally expensive and time consuming process, decreasing training time is valuable as it hastens the rate at which developers can iterate and improve upon the network.

For our CNN, we tested the differences between using ReLUs and neurons with saturating nonlinearities. Figure 4 displays the relative differences in average training loss over step number on a CNN. ReLUs had a more noticeable relative drop in average training loss for the first 2000 steps. After that, the difference became less noticeable. Regardless, we opted to utilize ReLUs in our CNN as the initial boost in training speed overall sped up the training process, ultimately providing a superior classifier after n steps.

3.3 Dataset

We trained and tested our CNN using mammogram data from the mini Mammogram Image Analysis Society (MIAS) [12]. The dataset consists of 332 grayscale classified mammograms with dimensions 1024×1024. It includes 209 normal, 61 benign, and 66 malignant instances. We had originally intended to use the Digital Database for Screening Mammography (DDSM) [13], which includes over 2000 instances with equal distribution of classifications, but we encountered unreasonably difficult obstacles when attempting to decompress the archaic

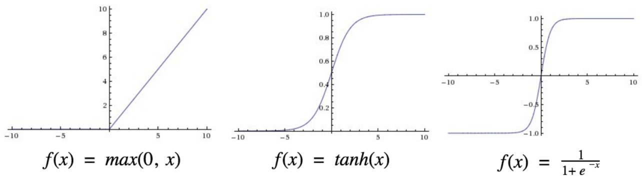

Figure 5: Saturating and non-saturating nonlinearities

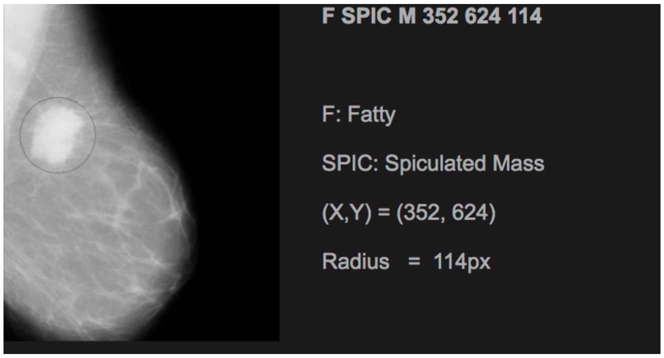

Figure 6: Example malignant data point

LPNG-format its images were stored in.

Certain characteristics of this dataset were suboptimal given the problem we aimed to solve. The first is that the sample size is not large enough to conclude with high statistical significance that this classifier will work for all mammograms. Additionally, these mammograms were not optimized for direct usage and contained black spaces from attempting to make each image 1024×1024 as well as artifacts from the initial screening. We discuss the pre-processing techniques we employed to counteract these setbacks in the section below.

The mini-MIAS dataset came with useful information for each instance in its dataset. Its data is labelled with the type of abnormality present like calcifications, masses, and asymmetries. Another helpful feature of the dataset that proved important in our final methods is the information included on the location of abnormalities. For benign and malignant cases, MIAS lists the X and Y coordinates and radius of the circle that most accurately encircles the location of the abnormality.

3.4 Image Preprocessing

A significant portion of time was spent on modifying our input to optimize classifier performance and computation time. The raw data is noisy, not centered directly on the abnormalities, and generally unfit to run through a classifier without preprocessing. A lack of powerful hardware to keep up with the complexity of our network forced us to shrink our input images to a size that could be processed in a reasonable amount of time.

In order to perform feature extraction, we subsampled each abnormal datapoint by extracting the abnormal region of interest. This was possible because each abnormal image was labeled with the x and y coordinates of the center of the abnormality. After cropping out the abnormalities, we scale the crops into to a uniform dimension of 48×48. In order to subsample normal mammograms, we choose a random spot in the center of the mammogram and crop out a 48×48 image. We finally parse through the subsampled data for poorly cropped subsamples, such as those that contained too much black space. From a performance standpoint, subsampling reduced the training time dramatically as it is far easier to train on a smaller image. Subsampling also provided us with an initial dataset that can allow a CNN to learn differences between benign, malignant, and normal instances.

Another technique we implemented during preprocessing was augmentation of the subsampled dataset using image transformations. For each subsample, we add to the dataset the subsample rotated by 90, 180, and 270 degrees. We also add horizontally reflected versions of each variation. This effectively allowed us to increase our sample size and increase generalization. We perform this technique with the assumption that transformed instances still accurately represent real abnormalities. Increasing our dataset size helps us fight overfitting and makes it possible to train classifiers based on one abnormality type.

Another problem we fixed during preprocessing was the uneven distribution of classes. For example, a higher number of normals may inflate the accuracy if the classifier consistently guesses normals correctly while still missing cancerous test points. In the early stages of our development phase, when our classifier was far from complete and had numerous bugs, we were still able to yield a nearly 70% accuracy. This is because nearly 2/3 of our dataset was comprised of normal cases, and simply guessing normal led to a decent accuracy. To better guarantee that our metrics were actually representative of the performance of our classifier, we randomly removed class instances until we attained an even distribution.

3.5 TensorFlow

Our Machine Learning framework of choice was TensorFlow version 0.7.1. TensorFlow is Google's Open Source Software Library for Machine Intelligence [11]. It was originally used exclusively in-house by the Google Brain team, Google's machine intelligence research organization, but was made open source in November of 2015 as a part of their open source initiative. The package is characterized by performing numerical computations with data flow graphs. Multi-dimensional arrays, or tensors represent the nodes and edges of the graphs in order to perform mathematical operations.

The actual library we used to build our CNN was the SciKit wrapper for TensorFlow called Scikit Flow. The reason we chose to use this library was the extent to which it abstracts many of the functions required to perform convolutions, update network weights based on error, and test batches of input efficiently. Also, Skflow minimizes the amount of configuration and new syntax needed to start building the CNN and decreases the learning curve to get started. Behind the scenes, a graph is built from the input structures specified and TensorFlow starts a session to run on a target device, giving more control over performance. From TensorFlow, we gain an easy and efficient way to test large sets of data on many different CNN configurations as well as a well-established library of machine learning functions and metrics.

3.6 Hardware

For testing and training our CNN, we used our personal Macbook Air and Pro laptops using the CPU version of Tensorflow. The Macbook Pro performed better due to its more powerful dual core 2.7Ghz Intel i5 processor. For optimal training performance, GPGPUs with SLI can provide highly parallelized computation that can greatly expedite training. Dual GPU processing was used in Krizhevsky et al [12].

4 Results

4.1 Analysis Methodology

Before presenting our classifier results, it is important to acknowledge which statistics are more important provided the domain space. For example, when testing detection on normal data, calcifications, and masses, recall score is more important than accuracy. This is because the main purpose of the classifier is to identify malignant abnormalities. From a statistical perspective, this means that we focus on minimizing false negatives. From a medical perspective, missing a malignant case translates to the kind of misdiagnosis that may prove to be lethal for a patient. Another statistic is the distribution of types of samples. Our dataset contained far more normal samples than either benign or malignant. In general, classifiers trained and tested on datasets with this skewed distribution have less informative accuracies since they are inflated by the number of normal test cases correctly identified. In practice, these classifiers may perform poorly when identifying malignant abnormalities due to low recall rates.

We built our testing and training datasets by shuffling the data and partitioning the data into 7 parts training and 1 part testing. We also ensure that the distribution of classes is even within each partition. We aimed to minimize overfitting by random shuffling and then guaranteeing even class distribution.

4.2 Detection and Classification

When training and testing on a shuffled dataset with an even distribution of normal, benign, and malignant data points, we observe the best recall and precision on a CNN with a configuration with learning rate of 0.002, one convolutional layer using 32 kernel filters of size 5×5, one convolutional layer using 64 kernel filters of size 5×5, 2×2 max pooling after each convolution, a dropout rate of 0.5, a fully connected hidden layer of 1024 neurons, and ReLUs. When trained over 5000 steps, we observe a recall rate of 59.612% for malignant values, general accuracy rate of 60.90%, and malignant precision rate 59.05%. Consequently, our best classifier managed to correctly classify 59.612% of all malignant test points.

We found the classifier struggled most when presented with data of every class. Sifting data by abnormality simplifies the task presented to the classifier. It is unsurprising we observe inferior results with all classes as it asks the classifier to do both detection of abnormalities as well as classification as benign or malignant. We also collect a training accuracy rate to help indicate overfitting. The training accuracy was 63.37%, which does not differ much from the test accuracy and does not strongly signal severe overfitting.

4.3 Calcifications Only

When training and testing on an even shuffled data of evenly distributed calcification abnormalities, we obtained the best results using the same architecture as mentioned in section 4.2 except we use a learning rate of 0.003 over 10,000 training steps. We observed recall accuracy of 100%, test accuracy of 100%, training accuracy of 95.03%, precision of 100%. The classifier performs best when only responsible for calcifications relative to classifiers trained on different datasets. Despite the perfect recall and test accuracies, it is evident through the training accuracy that the classifier is not perfect.

The favorable results indicate that there probably exist underlying features that indicate malignant versus benign calcification abnormalities. However, it is unclear what this feature is. The feature may be the spacing of the calcifications, sizes of the individual calcifications, specific patterns, etc. We must also address that the classifier may potentially generalize poorly as we trained the classifier off a relative small dataset. The total size of the augmented dataset is 160, which is relatively small and was derived from an even smaller initial dataset. Further, our lack of domain knowledge prevents us from accurately gauging if the microcalcifications presented in the dataset offers a sample that can be generalized to all microcalcifications. When judging

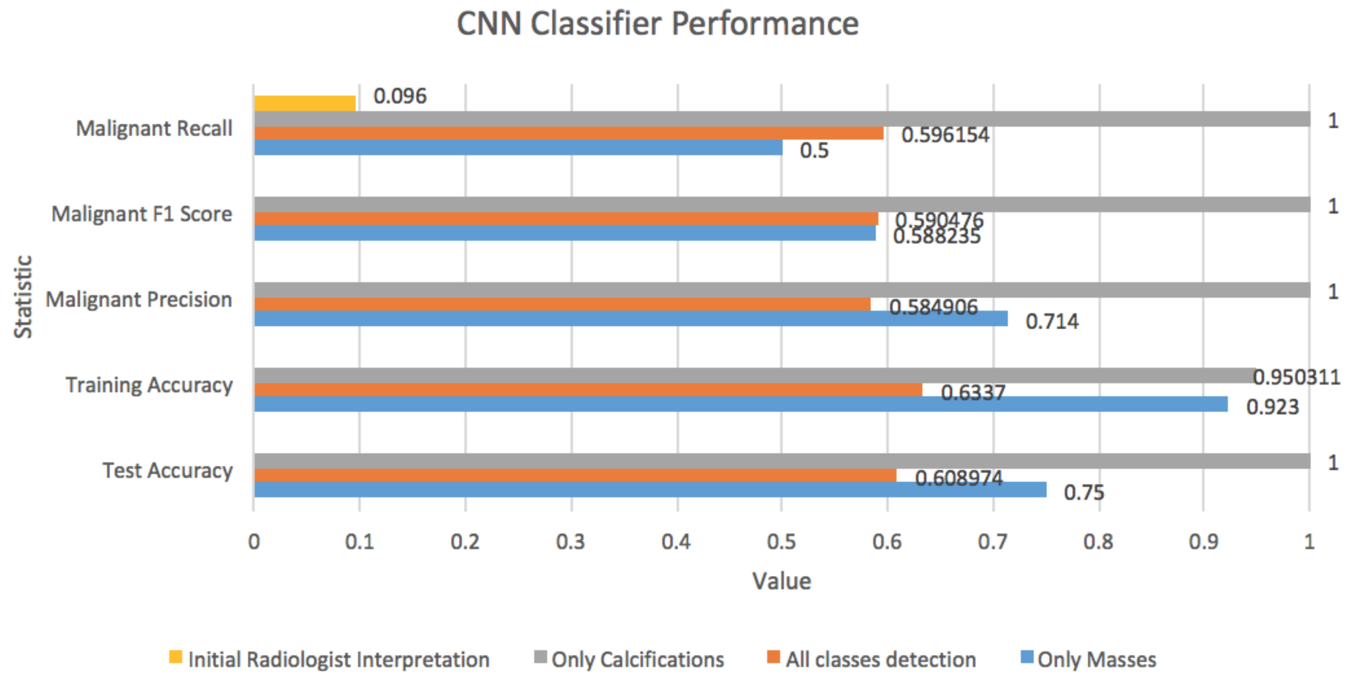

Figure 7: Various performance measurements for different classifiers

the performance of the classifier in this scenario, we successfully classify malignant cases of microcalcifications.

4.4 Masses Only

We obtain the best results when classifying only mass abnormalities using the same architecture as we used for only calcifications. When trained over 10,000 steps, we observed a malignant recall rate of 50.00%, an accuracy rate of 75.00%, a training accuracy of 92.30%, malignant precision of 71.40%, and a malignant F1 score of 50.00%. Our CNN seems to be relatively worse at differentiating malignant and benign masses compared to microcalcifications. The recall rate does not indicate that the classifier is well suited to classifying malignant cases. Another notable result is that the training accuracy is much higher than the test accuracy, signalling that the classifier does not generalize well and is likely overfitting to the training data.

One reason that the mass classifier exhibits poorer results may be the size of the dataset. The masses only data set is much larger than the calcification dataset and consists of 448 data points. Consequently, the mass only classifier is likely more indicative of actual classification success than the calcification only classifier. In spite of the more general dataset, the classifier still suffers from overfitting as the training accuracy far exceeds the test accuracy. The CNN trained on all data performed better than the mass classifier with a recall rate of 50.00%.

5 Discussion

When deciding on the neural network configuration, we found that choosing the perfect settings of variables was not easy due to the sheer number of combinations of factors that can be tested. Choosing the optimal values for kernel size, kernel value, learning rate, and many more factors can be a relatively nebulous process due to domain specific features and long training times. One CNN that works well for a specific domain may not work for another depending on what features one wishes to extract.

As mentioned earlier, our classifier is sometimes trained off a relatively small dataset that may not offer a rich selection of different abnormalities. We were also limited in this regard due to lack of medical knowledge regarding mammograms. Attempting to classify mammograms elucidated to us the extreme importance of domain specific knowledge when it comes to machine learning. A successful machine learning project requires some amount of domain specific expertise.

Despite the obvious importance of classifying whether a mammogram contains something potentially harmful to a patient, another important purpose of our classifier could be to detect benign abnormalities as well. While the need in the medical community for classification of malignancy might be more urgent, the speedup in processing that can come from detecting benign abnormalities computationally is still useful.

While we found in many papers that the ROC curve is used to account for unbalanced class distributions, we had difficulties using our library to generate the ROC curve. To account for skewed class distributions, resizing the input data set to include equal numbers of each class produced better results than a skewed distribution as expected and an ROC curve was not needed.

Lastly, we were heavily limited in the number of CNN parameter combinations we could benchmark because training the CNN was generally slow. Similarly, it would have also been extremely computationally intensive to perform k-fold cross-validation during testing. We attempted to install TensorFlow onto the Tufts University Computer Science Departments servers, but we were blocked by a slew of compatibility issues. Ultimately fewer CNN designs were attempted and cross-validation was rejected in favor of our single sample method.

6 Future Work

In the future more layers of convolution and more connected layers could be attempted using more powerful hardware. The number of tests we could do to find the optimal configuration was limited by the hardware that we used. For this reason many of our tests did not use large networks. We were also restricted by computational power in the number of iterations we could run on any one test, leading to fluctuations in accuracy between tests.

7 Conclusion

Despite the restriction of a small, noisy dataset, our classifier's ability to overperform human accuracy under certain settings indicates its potential viability in real-world application. As can be seen in Figure 7, the classifier significantly increases in performance when the type of abnormality is known. This means that the doctor is better off leaving the diagnosis to the classifier if they are able to identify the kind of abnormality with a high degree of confidence.

It is also likely that doctors will be able to obtain better training data. The mammograms they have access to are likely to be cleaner and better standardized than the ones available for public use. Doctors also have the resources to incorporate characteristics of the patient corresponding to each given mammogram, such as age or medical history, to give the classifier more features to work with. If there are not enough instances available to the doctors hospital or clinic to sufficiently train the classifier, we can imagine the possibility of various doctors crowdsourcing mammogram data.

The only concern for the classifiers effectiveness in practice is its recall rate. Due to the major repercussions of false negatives, it would be wise for doctors to perform biopsies on the patients that the doctor is suspicious of and the classifier predicts to be normal. Since the classifier struggles to attain higher than 60% recall (unless we stratify by calcifications), then the doctor would on average have to perform a biopsy 40% of the time. Nevertheless, the classifier proves to be significantly more efficient than human detection of cancer on mammograms, with a roughly six times higher recall.

8 References

[1] Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., & LeCun, Y. (2009, September). What is the best multi-stage architecture for object recognition?. In Computer Vision, 2009 IEEE 12th International Conference on (pp. 2146-2153). IEEE.

[2] Karpathy, A. (n.d.). CS231n Convolutional Neural Networks for Visual Recognition. Retrieved May 03, 2016, from = http://cs231n.github.io/convolutional-networks/

[3] Khosla, A., Raju, A. S., Torralba, A., & Oliva, A. (2015). Understanding and predicting image memorability at a large scale. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2390-2398).

[4] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[5] Lo, S. B., Chan, H., Lin, J., Li, H., Freedman, M. T., & Mun, S. K. (1995). Artificial convolution neural network for medical image pattern recognition. Neural Networks, 8(7-8), 1201-1214. doi:10.1016/0893-6080(95)00061-5

[6] Planey, K. (2011). Machine Learning Approaches to Breast Cancer Diagnosis and Treatment Response Prediction.

[7] Schell, M. J., Yankaskas, B. C., Ballard-Barbash, R., Qaqish, B. F., Barlow, W. E., Rosenberg, R. D., & Smith-Bindman, R. (2007). Evidence-based Target Recall Rates for Screening Mammography 1. Radiology, 243(3), 681-689. doi:10.1148/radiol.2433060372

[8] U.S. Breast Cancer Statistics. Breastcancer.org. (2016, March 2). Retrieved May 04, 2016, from = http://www.breastcancer.org/symptoms/understand_bc/statistics

[9] Wei, D., Chan, H., Helvie, M. A., Sahiner, B., Petrick, N., Adler, D. D., & Goodsitt, M. M. (1996). Classification of Mass and Normal Breast Tissue: A Convolution Neural Network Classifier with Spatial Domain and Texture Images. IEEE Transactions on Medical Imaging, 15(5).

[10] Breast Cancer Screening Statistics. = http://breastscreening.cancer.gov/statistics/performance/screening /2009/perf_age.html#f2

[11] Google Tensorflow. `http://tensorflow.org`

[12] J Suckling et al (1994): The Mammographic Image Analysis Society Digital Mammogram Database Exerpta Med-

ica. International Congress Series 1069 pp375-378.

[13] Michael Heath, Kevin Bowyer, Daniel Kopans, Richard Moore and W. Philip Kegelmeyer, in Proceedings of the Fifth International Workshop on Digital Mammography, M.J. Yaffe, ed., 212-218, Medical Physics Publishing, 2001. ISBN 1-930524-00-5.

[14] Convolution Effect on Image. = http://timdettmers.com/2015/03/26/convolution-deep-learning/

[15] Convolution Visualization. = https://docs.gimp.org/en/plug-in-convmatrix.html

[16] Max Pooling Visualization. = http://cs231n.github.io/assets/cnn/maxpool.jpeg