CS231N Project: Breast Cancer Tumor Detection

Ann Wu
Stanford University
yuanchih@stanford.edu

Takara Truong
Stanford University
takaraet@stanford.edu

Abstract

We implemented a breast cancer detection convolutional neural network (CNN), giving it ultrasound images of breast tumors as input and receiving per-image labels of "benign" or malignant" as output. We surveyed multiple literature sources both in the area of breast cancer detection as well as image segmentation as a general field during this process, and ultimately used a Mask R-CNN as our network architecture, which we manually implemented via PyTorch. For the dataset, we utilized the Breast Ultrasound Images Dataset [1] as made publicly available on Kaggle. We experimented with various components of our architecture, including the Region Proposal Network (RPN) architecture and (hyper)parameter values, and achieved an 45.24% training accuracy, 48.27% validation accuracy, and 43.41% test accuracy on our best-performing model.

1. Introduction

Application of artificial intelligence in the medical field has been a subject of interest for many, finding applications from smart hospitals with AI-driven patient care to disease detection. For our project scope we've chosen to focus on the specialization of disease detection, specifically cancer detection. From our literature survey, cancer detection via machine learning has been applied in nearly all fields, from lung cancer [13] to melanoma [8], and often with an accuracy that has met Bayes error (where Bayes error is synonymous with the diagnoses of professional radiologists). For our project, we've chosen to focus on breast cancer, specifically using a CNN as our neural network, giving the network mammography images as inputs, and having the network label each image as either having "benign" or "malignant" tumors.

2. Related Work

2.1. Traditional ML Methods For Breast Cancer Detection

We first took a look at the traditional (i.e. non-deep learning) methods for AI-driven breast cancer diagnostics. Popular algorithms for this line of research included Support Vector Machine (SVM), Random Forest (RF), and Bayesian Networks (BN). One such research paper [3] described these methods deployed on the Breast Cancer Wisconsin Diagnostics Dataset (which notably is not an image dataset but rather a numerical dataset), where the authors were able to get an average of 97.0% on SVM, 96.6% on RF, and 97.1% on BN.

Another traditional method was k-means clustering as deployed on the Wisconsin Dataset as well in 2016 [2], which evaluates the impact using the dataset's centroid and distance measures. This approach obtained approximately 92% average position prediction accuracy.

A third traditional method of note is utilizing feature discovery and classification along with decision trees, as experimented in 2011 [5], which utilized data preprocessing techniques such as reduction, attribute selection, and data cleaning. This method along with the main classification method of decision trees yielded approximately 94% accuracy on the Wisconsin Dataset [14]. While the datasets used in these traditional approaches were not images [14], we considered this survey important to gain understanding of what features ended up being salient for accurate predictions.

2.2. Deep Learning Methods For Breast Cancer Detection: Famous Networks

We moved on to explore deep learning breast cancer diagnostics techniques, particularly those that utilized images as inputs. End-to-end deep learning training with an all-convolutional network started becoming more popular in this field, due to its ability to extract complex features from the mammography images.

We first took a look at how several famous CNN architectures performed in this space, specifically GoogLeNet,

VGGNet, and ResNet. This paper [10] shows a survey across these three architectures with the following results: GoogLeNet achieved 93.5% classification accuracy, VGGNet achieved 94.15%, and ResNet achieved 94.35%.

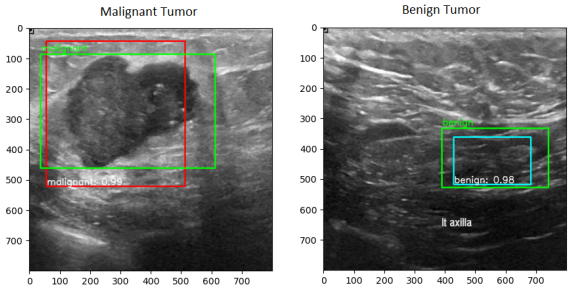

Figure 1. Example classifications of a malignant tumor (right) and benign tumor (left) from our model. Green is for ground truth, Red is for a malignant tumor classification, and blue is for a benign tumor [15]

2.3. Deep Learning For Breast Cancer Detection: Regions of Interest

Many of the relevant difficulties in breast cancer detection in mammography images were discussed [9], such as the typically small cancerous region of interest (ROI) compared to a full-field digital mammography image, as well as the fact that typical large mammography databases lack ROI annotations that are very labor and cost intensive to assemble.

This paper [9] discusses interesting strategic training to satisfy the need of a large training dataset with utilizing the largely-unannotated mammography databases. One common strategy that they tested is using a classifier in sliding window fashion to identify local patches and generate a probabilistic grid. They also experimented with CNN architectures to yield higher accuracies, such as using convolutional layers as top layers to preserve spatial information. Ultimately, they reported that against the average sensitivity of digital screening mammography in the U.S. of 86.9% and average specificity of 88.9%, the authors were able to achieve sensitivy of 86.7% and specificity of 96.1% with their deep learning model.

Another interesting paper [4] that has a similar approach utilized a two-stage network: an initial patch-based classification stage that separates the image into millions of small positive vs. negative patches for tumor identification, and a heatmap-based post-processing stage that generates probabilities that each patch contains tumor. This approach obtained an AUC of 0.925 of whole slide image classification and 0.7051 for tumor localization task, compared with a pathologist that obtained an AUC of 0.966 and 0.733, respectively.

We therefore observed that best-in-class methods tended to have a two-stage approach, an initial segmentation stage of the image into regions of interest, and a later identification stage that labels each region of interest with probability of existing tumor. This served as guidelines for our approach to our solution.

2.4. Survey on Deep Learning Approaches to Image Segmentation

As our findings led us towards having an image segmentation stage in our network, we elected to do a survey of state-of-the-art segmentation architectures. One such survey analyzed categories including fully convolutional networks, CNNs with graphical models, encoder-decoder based models, R-CNNs, recurrent neural networks, attention-based models, GANs, and more [12]. From this survey, we learned that R-CNN based models is a strong candidate for our network of choice, due to its extraction of regions of interest through a built-in region proposal network.

We therefore continued to learn about the Mask R-CNN architecture through the original paper [7], as well as the Fast R-CNN on which the Mask R-CNN was based [11], for a deeper understanding of their structure for our eventual implementation.

2.5. Survey on Machine Learning and Deep Learning Applications in Breast Cancer Diagnosis

As a final summary of related work exploration, we reviewed a comparative survey between traditional machine learning methods (as per [3]) and modern deep learning methods (as per [9]) on AI-driven breast cancer diagnostics, to determine their relative effectiveness [6].

The results show that deep learning methods outperform conventional machine learning methods for diagnosing breast cancer when the dataset is sufficiently large, thus serving as a motivator for this area of research.

3. Dataset

We surveyed several publicly available datasets and elected to use the Breast Ultrasound Images Dataset [1] available on Kaggle, due to its ease of use and the fact that bounding boxes can be extracted from the dataset to train our network.

3.1. Breast Ultrasound Images Dataset

This 2018 dataset contains PNG images of breast cancer ultrasound scans collected from women of ages between 25 and 75 years old, and consists of three class categories: normal, benign, and malignant. Each ultrasound image is accompanied by an mask image of identical size, comprised of white pixels within the bounded region of interest of the

tumor, and black pixels everywhere else; images of normal breasts (in other words, with no tumors at all) is hence entirely black pixels. An example of an image and its corresponding mask image is shown respectively in Fig 2 and Fig 3.

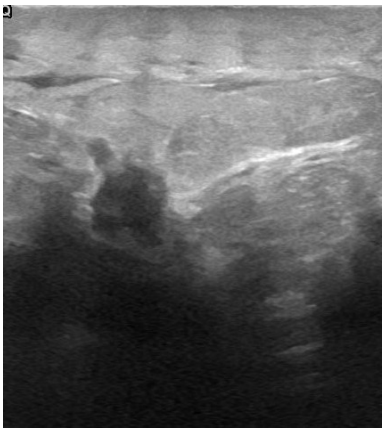

Figure 2. Malignant tumor ultrasound image: example 3 [1]

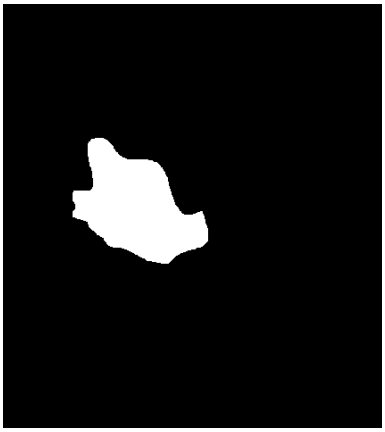

Figure 3. Malignant tumor ultrasound image: example 3's mask [1]

The number of patients scanned is 600 female patients, and the dataset consists of 780 images with average image size of 500x500 pixels.

category	num samples
benign	437
malignant	210
normal	133

3.2. Data Pre-Processing

Since we are using a network with a Region of Proposal Network (RPN), we need bounding boxes for each image input. Therefore, we wrote a script that extracts the xmin, xmax, ymin and ymax from the corresponding mask image of each ultrasound image (to define a rectangular bounding box for the tumor in the image), and wrote the coordinates into a metadata.csv along with the filename of the PNG file as well as the corresponding category label.

We also decided to keep the classification binary, between benign and malignant tumors, and hence eliminated the 'normal' category from our experimental data.

3.3. Training and Testing Split

Once metadata.csv was created, we can then shuffle the data and split into training and testing/validation sets. We utilized the Fisher-Yates shuffle on metadata.csv to shuffle the rows, and used a 90/5/5 split for training/testing to split the shuffled data into separate CSV files to be sourced in the relevant parts of the network pipeline. This resulted in 597 training samples, 34 validation samples, and 34 testing samples.

4. Method

While we had originally planned to use a Mask R-CNN for this project, due to time constraint we opted to use a Faster R-CNN, which is a very similar algorithm that is the direct predecessor of the Mask R-CNN algorithm. Since this is our first exposure to computer vision (and want to gain better understanding through implementation), we aimed to develop the algorithm from scratch resulting in approximately 600 lines of code.

4.1. Faster R-CNN

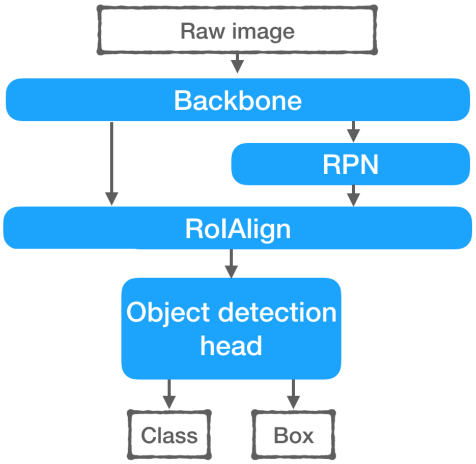

Figure 4. Faster R-CNN architecture [15]

There are four main components to implement for the Faster R-CNN (Fig 4). These are: Backbone, Region Proposal Network, Region of Interest Alignment, and Object Detection Head.

4.1.1 Backbone

The backbone model serves to map images to an embedding space which is used throughout other parts of the model. For this component, we elected to use a VGG-16 network where the embedding is extracted before the final linear layers. All input images are resized to 3x800x800 before being fed into the backbone; at the output of the backbone, we get a 512x50x50 embedding of the resized image.

4.1.2 Region Proposal Network

The RPN is a neural network that finds the regions of interests (ROIs); in other words, objects within the image to identify. This requires two key actions: find how much to resize predefined anchors (otherwise called bounding boxes) to better overlap with objects in the image (we defined 22500 anchors to adjust as a recommended value), and determine whether or not there exists an object within that bounding box (also referred to as determining foreground vs. background objects).

Given the many overlapping bounding boxes, we use non-max suppression to filters proposals based on the likelihood of an existing object. As a result of the non-max suppression, we go from 22500 anchors to a max of 2000 anchor proposals. From this reduced subset, we sample anchors that have strong probability of containing either foreground or background objects to move on to the next stage.

This is arguably the part of the network on which we spent the most time, as accurate predictions can only follow if the regions of interest are first correctly identified by the network. It accounts for more than 50% of the written code.

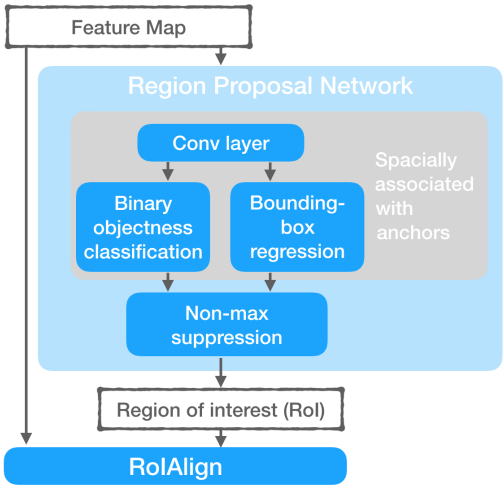

Figure 5. Faster R-CNN architecture [15]

4.1.3 Region of Interest Alignment

We additionally applied ROI alignment so that we align the output cell boundaries of the ROI maps to the input feature map grid. Taking this step has historically been shown to improve model accuracy, so we elected to add this component to our network.

We also implemented ROI pooling as an experiment; however, initial results were evidently worse, so we kept the ROI alignment version.

4.1.4 Object Detection Head

We need a final layer that predicts the class label of the ROIs defined by the previous layers as well as final adjustments to the ROI's. For this we simply used fully connected layers to reduce dimensionality down to a softmax layer to ultimately predict the label probability.

4.2. Loss Functions

Our loss function scheme is to aggregate two loss functions: one for object classification and another for bounding box regression. The object classification is simple cross entropy loss as defined in class, which we define here again:

$$L = -\frac{1}{m}\sum_{i}^{m} y_i * log\hat{y_i} + (1 - y_i) * log(1 - \hat{y_i})$$

The second loss is taken from the original Faster R-CNN paper [11], which is defined as such:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}}\sum_{i} L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_{i} p_i^* L_{reg}(t_i, t_i^*)$$

These two losses are added together based on a simple weighting scheme.

This scheme is used in both the RPN and the object detection head to produce one loss value per RPN and head. Lastly, both RPN loss and object detection head loss are added together before being backpropagated.

4.3. Algorithmic Validation

To validate that the model and algorithm works, we overfit to a handful of examples. We first validated the RPN by taking a single example and seeing if the RPN was able to identify regions where there might be a tumor. After this worked, we did the same process on the entire model. Throughout this process we tuned hyperparameters such as the learning rate to speed up convergence, and experimented with using different optimizers such as SGD with momentum and Adam to see which decreased the loss the quickest. We ended up using learning rate of 0.0001 and using Adam as our optimizer.

5. Experiments

5.1. RPN Architecture

In the standard Faster R-CNN architecture, there are two places that adjust the proposed bounding boxes. One lies in the region proposal network, and another in the object detection head layer. If the region proposal network provides a sufficiently good bounding box, then the object detection head layer should not need to adjust the bounding box further. Given this, we perform an ablation study specifically looking at the effect of adding the second bounding box adjustment. Both RPNs took approximately 13000 epochs over the entire training dataset to converge to their final training accuracy, though with different accuracy results as shown in the Results and Evaluation section.

6. Results and Evaluation

6.1. Accuracy Results Relative to RPN Architecture

From our method, we have two possible model solutions: one with the bounding box adjustment in the head component, and one without. These two solutions report different accuracy across all three datasets. To arbitrate between our two models, we used the scoring mechanism as follows that takes as input the three accuracy values per model (which prioritizes validation and testing accuracy over training accuracy):

$$score = \frac{train + 2 * val + 2 * test}{5}$$

Model	Train Acc	Val Acc	Test Acc	Score
With bbox adj	50.13%	43.75%	41.16%	**43.99**
W/o bbox adj	45.24%	48.27%	43.41%	**45.72**

Table 1. Removing the second bounding box adjustment results in a better scoring model, due to accuracy improvements for validation and test.

Given the same amount of training iterations (approximately 13000 epochs over the entire training set), it was interesting to see that not using a second bounding box adjustment gives worse training accuracy but better validation and testing accuracy by approximately 5%.

One possible explanation is that the first model is overfitting the training examples, and perhaps adding regularization would be beneficial.

6.2. Failure Modes

We inspected the mislabeled data to see where the network could have made mistakes in identifying it.

6.2.1 Data Quality

We noticed that similar looking objects have different bounding box classifications. For example in Fig 6, we see two images of malignant tumors displayed that both have a dark, similarly-shaped and sized object in the lower-middle section. However, only the right image labels that object as the tumor with its ground truth bounding box; in the left image, the ground truth bounding box labels a wholly different region as the tumor. Further inspection of other failures in the dataset shows this phenomenon appear again and again.

We can therefore hypothesize that the network is in fact learning to generate regions of interest that correspond to the training set; divergent and hence confusing results is resulting in the lower-than-expected accuracy. We wonder if the quality of the dataset is contributing to this problem.

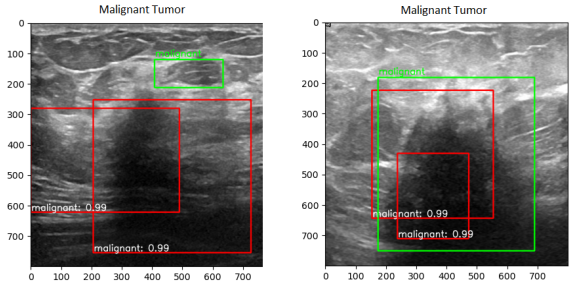

Figure 6. The test set shows confusing malignant ground truths.

6.2.2 Multiple Classifications on the Same Ground Truth

Another interesting phenomenon we found is that multiple classifications on the same ground truth lowers the reported accuracy. Our method discourages multiple classifications on the same ground truth: in Fig 7, we see that the benign example on the left shows, there are two correct classifications and one ground truth. The resulting network attributes to this example an accuracy of 1/2. Similarly for the malignant example to the right, there are three correct classifications but only one ground truth, and the network attributes to this example an accuracy of 1/3.

6.3. Conclusions and Future Work

In this project, we implemented a Faster R-CNN by hand that takes in breast cancer ultrasound images and their corresponding bounding box coordinates, and outputs image labels of either ”benign” or ”malignant”. Our final model, which comprises of the standard Faster R-CNN implementation but without the bounding box adjustment in the head component, reports a training set accuracy of 45.24%, validation set accuracy of 48.27%, and test set accuracy of

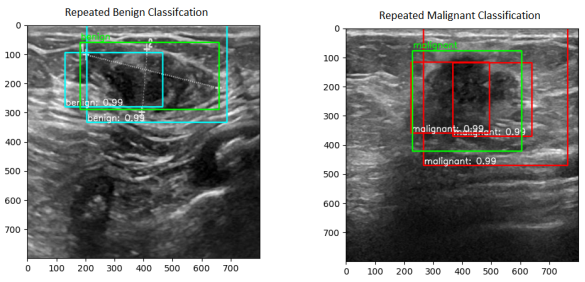

Figure 7. Repeated classifications lowers reported accuracy. Benign example (left), malignant example (right).

43.41%. Upon inspection of the failed labels, we noticed both inconsistent bounding box definitions in the original dataset (i.e. noisy data) as well as implementation non-idealities that, if fixed, could likely lead to better accuracy results.

Given more time and compute, we would have liked to try a few things. We would have liked to spent more time and compute sweeping some of the parameters and hyper-parameters, such as the anchor box sizes and RPN proposal counts. We would have liked to run experiments with better quality datasets rather than the one that we used in the training and testing of our network. Given more time, we would have also liked to experiment more with the architecture of the Faster R-CNN network components (such as the backbone architecture, the layer types and number of layers of the head layer, and more) to gain more insight into how they affect network behavior and performance.

Nevertheless, we learned a great deal about the various flavors of R-CNNs and implementing them from scratch, as well as the complexities of working with breast cancer mammography images!

6.4. Contributions and Acknowledgements

Both members contributed equally to the coding and written portions of the project. We would like to thank the teaching staff for a great course!

References

[1] Khaled H Fahmy A. Al-Dhabyani W, Gomaa M. Dataset of breast ultrasound images. https://www.kaggle.com/datasets/aryashah2k/breast-ultrasound-images-dataset. 1, 2, 3

[2] Umesh Gupta Sonal Jain Ashutosh Kumar Dubey. Analysis of k-means clustering approach on the breast cancer wisconsin dataset. 1

[3] Dana Bazazeh and Raed Shubair. Comparative study of machine learning algorithms for breast cancer detection and diagnosis. https://ieeexplore.ieee.org/abstract/document/7818560. 1, 2

[4] Rishab Gargeya Humayun Irshad Andrew H Beck Daeyong Wang, Aditya Khosla. Deep learning for identifying metastatic breast cancer. 2

[5] Dr.K.Usha Rani D.Lavanya. Analysis of feature selection with classification: breast cancer datasets. 1

[6] Shailender Kumar Nanhay Singh Gunjan Chugh. Survey on machine learning and deep learning applications in breast cancer diagnosis. 2

[7] Piotr Dollár Ross Girshick Kaiming He, Georgia Gkioxari. Mask r-cnn. 2

[8] Megan Lewis. An artificial intelligence tool that can help detect melanoma. https://news.mit.edu/2021/artificial-intelligence-tool-can-help-detect-melanoma-0402. 1

[9] Joseph H. Rothstein Eugene Fluder Russell McBride Weiva Sieh Li Shen, Laurie R. Margolies. Deep learning to improve breast cancer detection on screening mammography. *Nature*. 2

[10] ZahoorJan Ikram Ud Din Joel J. P.C Rodrigues SanaUllah Khan, NaveedIslam. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. 2

[11] Ross Girshick Jian Sun Shaoqing Ren, Kaiming He. Faster r-cnn: Towards real-time object detection with region proposal networks. 2, 4

[12] Fatih Porikli Antonio Plaza Nasser Kehtarnavaz Shervin Minaee, Yuri Boykov and Demetri Terzopoulos. Image segmentation using deep learning: A survey. 2

[13] Elizabeth Svoboda. Artificial intelligence is improving the detection of lung cancer. https://www.nature.com/articles/d41586-020-03157-9, 2020. Online; accessed 07-May-2022. 1

[14] William Wolberg. [uci] breast cancer wisconsin diagnostic dataset. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29, 2016. Online; accessed 07-May-2022. 1

[15] Xiang Zhang. Understanding mask r-cnn basic architecture. https://www.shuffleai.blog/blog/Understanding_Mask_R-CNN_Basic_Architecture.html. 2, 3, 4