

UNIVERSITY OF THE PHILIPPINES MANILA  
COLLEGE OF ARTS AND SCIENCES  
DEPARTMENT OF PHYSICAL SCIENCES AND MATHEMATICS

MAMMO: AN APPLICATION OF CONVOLUTIONAL NEURAL  
NETWORKS ON BREAST CANCER SCREENING

A special problem in partial fulfillment  
of the requirements for the degree of  
**Bachelor of Science in Computer Science**

Submitted by:

Sigfreed John S. Angeles

June 2018

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

## ACCEPTANCE SHEET

The Special Problem entitled “Mammo: An Application of Convolutional Neural Networks on Breast Cancer Screening” prepared and submitted by Sigfried John S. Angeles in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

---

**Vincent Peter C. Magboo, M.D., M.Sc.**  
Adviser

**EXAMINERS:**

	<b>Approved</b>	<b>Disapproved</b>
1. Gregorio B. Baes, Ph.D. ( <i>cand.</i> )	_____	_____
2. Avegail D. Carpio, M.S.	_____	_____
3. Richard Bryann L. Chua, Ph.D. ( <i>cand.</i> )	_____	_____
4. Perlita E. Gasmen, M.S. ( <i>cand.</i> )	_____	_____
5. Marvin John C. Ignacio, M.S. ( <i>cand.</i> )	_____	_____
6. Ma. Sheila A. Magboo, M.S.	_____	_____
7. Geoffrey A. Solano, Ph.D. ( <i>cand.</i> )	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

---

**Ma. Sheila A. Magboo, M.Sc.**  
Unit Head  
Mathematical and Computing Sciences Unit  
Department of Physical Sciences  
and Mathematics

---

**Marcelina B. Lirazan, Ph.D.**  
Chair  
Department of Physical Sciences  
and Mathematics

---

**Leonardo R. Estacio Jr., Ph.D.**  
Dean  
College of Arts and Sciences

## Abstract

In screening mammography for breast cancer, radiologists often identify the region of interest (ROI) in a mammogram, and analyze it whether it's benign or malignant, and whether it's a mass or calcification abnormality, for the course of treatment is dependent on the result of the exam. However, there are cases where the ROI is extremely difficult to classify, and radiologists sometimes have conflicting readings. Consequently, Mammo, an application that utilized Convolutional Neural Networks (CNN), was created in this study providing radiologists and medical field students/trainees a tool for classifying their mammograms as well as an exam for the students/trainees. The CNN architecture used was ResNet-50, a CNN where layers can be stacked without the typical vanishing or exploding gradients problem that plague CNNs when more layers are stacked. Using CBIS-DDSM (Curated Breast Imaging Subset of the Digital Database for Screening Mammography), initially, the network achieved an accuracy rate of 33%. This is caused by lack of data augmenting technique performed on the dataset as well as an imbalance of instances per category. After balancing the instances, CNN ResNet-50 achieved an overall accuracy rate of 43%.

*Keywords:* breast cancer, mammography, convolutional neural network, resnet-50, mass, calcification, benign, malignant, data augmentation

# Contents

<b>Acceptance Sheet</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>List of Figures</b>	<b>v</b>
<b>I. Introduction</b>	<b>1</b>
A. Background of the Study . . . . .	1
B. Statement of the Problem . . . . .	4
C. Objectives of the Study . . . . .	4
1. Training Module . . . . .	4
2. User Objectives . . . . .	5
D. Significance of the Project . . . . .	6
E. Scope and Limitations . . . . .	6
F. Assumptions . . . . .	6
<b>II. Review of Related Literature</b>	<b>8</b>
<b>III. Theoretical Framework</b>	<b>14</b>
A. Breast Cancer . . . . .	14
B. Screening Mammography . . . . .	20
C. Machine Learning . . . . .	24
D. Convolutional Neural Networks . . . . .	26
E. Residual Neural Networks . . . . .	28
<b>IV. Design and Implementation</b>	<b>30</b>
A. System Overview . . . . .	30
B. Use Case Diagram . . . . .	31
C. Activity Diagram . . . . .	32
D. Technical Architecture . . . . .	34
E. Data Set . . . . .	34

<b>V.</b>	<b>Results</b>	<b>35</b>
1.	Preparing the Dataset . . . . .	35
2.	Setting the Hyperparameters . . . . .	35
3.	Improving the Accuracy . . . . .	36
A.	Mammogram Analysis . . . . .	37
1.	Uploading Mammograms . . . . .	37
2.	Editing and Deleting Mammograms . . . . .	37
3.	Analyzing the Mammograms . . . . .	38
B.	Quiz . . . . .	38
1.	Taking the Quiz . . . . .	38
2.	Quiz Results . . . . .	39
<b>VI.</b>	<b>Discussion</b>	<b>45</b>
<b>VII.</b>	<b>Conclusions</b>	<b>47</b>
<b>VIII.</b>	<b>Recommendations</b>	<b>48</b>
<b>IX.</b>	<b>Bibliography</b>	<b>49</b>
<b>X.</b>	<b>Appendix</b>	<b>56</b>
<b>XI.</b>	<b>Acknowledgement</b>	<b>66</b>

## List of Figures

1	A back to back position of the left and right breasts of both the MLO and CC views, Mammo . . . . .	20
2	The five different types of mammogram masses, Mammo . . . . .	21
3	The five different types of mammogram mass margins, Mammo . . . . .	21
4	The five different types of microcalcifications based on shape, Mammo . . .	22
5	An intuitive overview of a standard CNN, Mammo . . . . .	26
6	An illustration of an input volume (i.e. 32x32x3 input image) at the left and the first convolutional layer (a volume of neurons) to the right; image, Mammo	27
7	Average pooling vs max pooling, Mammo . . . . .	27
8	Left: a plain CNN with 34 parameter layers. Right: a RNN with 34 parameter layers. The dotted shortcuts increase dimensions, Mammo . . . . .	29
9	System Overview, Mammo . . . . .	30
10	Use Case Diagram, Mammo . . . . .	31
11	Radiologist Activity Diagram, Mammo . . . . .	32
12	Medical Field Student Activity Diagram, Mammo . . . . .	33
13	ResNet-50's hyperparameters, Mammo . . . . .	36
14	The log file of the model showing diminishing loss on the first few iterations as indicated by the red arrows, Mammo . . . . .	37
15	The log file of the model showing diminishing loss on the last remaining iterations as indicated by the red arrows, Mammo . . . . .	38
16	The log file of the model with the balanced dataset diminishing improving loss on the first few iterations as indicated by the red arrows, Mammo . . .	39
17	The log file of the model with the balanced dataset diminishing improving loss on the last remaining iterations as indicated by the red arrows, Mammo	40
18	The accuracy of the model after balancing the dataset, Mammo . . . . .	40
19	Mammo home page, Mammo . . . . .	40
20	Three uploaded mammograms, Mammo . . . . .	41
21	Editing metadata of the mammogram, Mammo . . . . .	41

22	Results of the analysis made by the CNN module part 1, Mammo . . . . .	42
23	Results of the analysis made by the CNN module part 2, Mammo . . . . .	42
24	A generated PDF file of Mammogram Analysis Report, Mammo . . . . .	43
25	A sample of the quiz, Mammo . . . . .	43
26	Submit the quiz, Mammo . . . . .	44
27	Results of the quiz, Mammo . . . . .	44

# I. Introduction

## A. Background of the Study

Breast cancer is an uncontrolled growth of breast cells. These cells are also known as malignant tumors. On the other hand, benign tumors are not considered cancerous because their cells are close to normal in appearance; they grow slowly; and they do not invade nearby tissues or spread to other parts of the body. On the other hand, malignant tumors are cancerous due to the cells spreading to distant areas of the body. [1].

To date, the precise causes of breast cancer are unknown. But there are known main risk factors attributed to breast cancer. Among the most significant factors are advancing age and a family background of breast cancer. Other factors include hormonal, lifestyle, and environmental factors. Still, most women considered at high risk for acquiring breast cancer do not get it, while many with no known risk factors do develop breast cancer [2].

Early symptoms and signs of breast cancer are usually unrecognizable. There are also some reports of larger breast cancer cases not producing symptoms and signs. When symptoms do occur, the most common symptom is a lump or mass in the breast or underarm area. Other symptoms include nipple discharge, nipple redness, newly inverted nipple, changes in the breast skin texture such as puckering or dimpling, and swelling of part of the breast [3].

As for the screening and diagnosis of breast cancer, mammography is the way to go. Mammography is a specialized medical imaging that uses a low-dose x-ray system to see inside the breasts. Screening mammography is imperative in early detection of breast cancers because it can show changes in the breast up to two years before a patient or physician can feel them. Furthermore, early detection of breast cancer is crucial because it is at that phase when they are most curable [4]. Other common methods of breast cancer screening include a clinical breast exam, breast magnetic resonance imaging (MRI), and breast ultrasound.

In this study, there are two common types of abnormalities that may be observed in



screening mammography namely calcification cases and mass cases. Breast calcifications are small spots of calcium salts [5] whereas masses are three-dimensional lesions. Given this, four classifications can be made from a region of interest (ROI) in any mammogram specifically benign mass, malignant mass, benign calcification, and malignant calcification. In mammogram analysis, round, oval, or lobulated masses with defined margins or borders are benign tumors. Moreover, larger calcifications with regular or oval shapes are benign, while smaller, irregular calcifications are potentially malignant.

Recently, there have been advances in breast cancer treatment technology providing more options for those afflicted. These options include surgery (mastectomy and lumpectomy), radiation therapy, hormonal (anti-estrogen) therapy, chemotherapy, and targeted therapy [6]. However, the kind of treatment to be recommended depends on the type of breast cancer, the size of the tumor, and the stage of the breast cancer [7].

In the United States alone, the estimated number of new cases for 2018 is 266,120 for both men and women with women being the majority (268,670). It is also estimated that 40,920 deaths will arise from this [8]. Moreover, the Philippines was declared the country with the most number of cases of breast cancer out of 197 countries in 2016 according to data released by Philippine Obstetrical and Gynecological Society [9]. Moreover, the Philippines has the highest incidence of breast cancer in Asia with one in every 13 Filipino women at risk of being afflicted with the disease [10].

The high incidence of breast cancer brings not only financial troubles to the affected families but economic repercussions to the country as well. In 2015, a news article reports about an action study, Asia Costs in Oncology, conducted across eight countries in Southeast Asia: Cambodia, Indonesia, Laos, Malaysia, Myanmar, Thailand, Vietnam, and the Philippines. The study aimed at assessing the impact of cancer on household economic wellbeing, patient survival, and quality of life. The results of the study showed that of the 9,513 patients followed up at 12 months after cancer diagnosis in SEA, over 75 percent of patients had the worst outcomes specifically 29% died of cancer while 48% experienced

financial catastrophe [11].

With the advent of machine learning, there have been several applications of machine learning algorithms specifically artificial neural networks (ANN) for data problems such as image processing, character recognition, and forecasting (i.e. predicting stock prices). Recently, the research on ANN has led to the development of a new form of machine learning specifically deep learning through convolutional neural networks (CNN) [12]. CNN has been developed by Szegedy et al., and is the most popular neural network architecture for deep learning for images. One advantage of CNNs compared to ANNs is the robustness of CNNs in image detection especially when dealing with distortions such as changes in shape due to camera lens, different lighting conditions, different poses, presence of partial occlusions, horizontal and vertical shifts, etc. This entails better generalization of a model. Another main advantage is that CNNs have significantly fewer parameters than ANNs, therefore memory requirements are drastically reduced. And, since the number of parameters would be much lower, training time is proportionately reduced [13].

Some recent applications of CNN for medical imaging include the diagnosis of Malaria through blood smear images [14], Thorax disease through chest x-ray images [15], and *Helicobacter pylori* gastritis through endoscopic images [16]. Moreover, there are also several CNN applications for medical imaging for breast cancer screening. Given this, an expert system with a CNN application for mammogram screening should be feasible.

In 2015, a Residual Neural Network (RNN), a type of CNN, called ResNet-152 developed by Microsoft won the ImageNet Large Scale Visual Recognition Competition (ILSVRC), where it topped with a top-5 error rate of 3.57% beating human-level performance on the ImageNet data set. Moreover, ResNet-152 outperformed other popular CNN architectures such as GoogleNet, VGG, and AlexNet by incorporating the residual learning framework. Residual learning allows deeper CNNs to be trained safely without the degradation problem that occurs with the addition of more layers. With deeper CNNs, more complex tasks can be solved, and classification/recognition accuracy also increases [17].

## **B. Statement of the Problem**

As stated, mammography is the recommended screening technique. Consequently, the mammogram is analyzed by the radiologist in order to check for lesions and to give a proper reading. However, there are cases where, even with the most experienced of radiologists, it is extremely difficult to identify and distinguish malignant tumors from benign ones. And it is also tedious for a radiologist to analyze a large batch of mammograms if such an event arises. Also, more often than not, students or trainees in medical fields would have a difficult time learning concepts and mechanics regarding mammogram screening, since there is a lack of online learning materials for screening mammography especially in the analysis and interpretation of mammograms.

## **C. Objectives of the Study**

### **1. Training Module**

This study aims to build an expert system called Mammo through the use of a pre-trained CNN and fine-tuning its parameters. The CNN model is slightly patterned off of Shen's pretrained ResNet-50, a RNN with 50 layers, patch classification model [18], since the author trained on the same data set used in this study, CBIS-DDSM (Curated Breast Imaging Subset of the Digital Database for Screening Mammography), and achieved a high accuracy of 0.89. However, the author's model trained on the whole image dataset whereas the model in this study was off of the cropped ROI image version of the dataset. The model was trained given the following details:

1. Train the network through the gradient descent by back propagation algorithm.
2. Train the network with the following parameters:
  - (a) Set the learning rate according to 0.001.
  - (b) Set the total number of epochs to 40.

3. Convert the mammograms from DICOM into PNG format in order for the mammogram images to be utilized by python libraries.
4. Relabeled the images in an iterative manner starting from 0 e.g. 0.png, 1.png, 2.png, etc.
5. Train the network to classify a ROI into four categories: benign mass, malignant mass, benign calcification, and malignant calcification.

## **2. User Objectives**

Mammo assists radiologists on the screening of mammograms specifically by providing a support tool to augment radiologist's interpretation particularly in difficult cases. Moreover, this study aims to provide a means for medical students to verify the diagnoses they've made especially when they're undertaking a course in screening mammography:

1. Allows the radiologist to
  - (a) Upload and input mammogram/batch of mammograms
  - (b) View the mammogram/batch of mammograms
  - (c) Output the classification whether there exists some benign mass, malignant mass, benign calcification, or malignant calcification
  - (d) Download and save the classification/s made into a PDF file
2. Allows students or trainees in medical fields to
  - (a) Upload and input a mammogram/batch of mammograms
  - (b) View the mammogram/batch of mammograms
  - (c) Output the classification whether there exists some benign mass, malignant mass, benign calcification, or malignant calcification
  - (d) Download and save the classification/s made into a PDF file
  - (e) Test their knowledge of mammograms through a multiple choice quiz

## **D. Significance of the Project**

The expert system assists radiologists by providing a second, unbiased opinion regarding the diagnosis which would improve radiologist's diagnostic confidence. The system may also act as an arbiter for conflicting readings.

The system can be used as a learning tool for medical students or trainees by providing them an avenue to verify their practice mammogram readings.

## **E. Scope and Limitations**

1. The system is implemented as a web application but would not be publicly distributed.
2. Only mammograms in PNG, JPG, and JPEG format and have a size of at least 1152x896 are accepted as input.
3. The network is only trained under CBIS-DDSM.
4. The network can only make 4 conclusions with regards to the diagnosis of the mammogram/s namely benign mass, malignant mass, benign calcification, and malignant calcification.
5. The exam given to the medical field student is only multiple choice, and its question pool is limited.
6. The system requires the user to specify a ROI from the mammogram for analysis instead of having the whole image analyzed.

## **F. Assumptions**

1. The mammograms that the user would input is assumed to be anonymized or permitted by the patient.
2. The mammogram can be of the left or right breasts and can be craniocaudal (CC) or mediolateral oblique (MLO).
3. The system does not cover other means of breast cancer medical imaging such as Magnetic Resonance Imaging (MRI) scans.

4. The user is assumed to have a built-in NVIDIA graphical processing unit (GPU) in his/her machine.

## II. Review of Related Literature

In recent years, there have been an influx of deep learning applications from various researchers for medical imaging tasks. To name a few, Penas et. al [14] implemented a CNN to recognize Malaria parasites on microscopic images of thin blood smears; Shichijo et. al [16] explored the role of artificial intelligence in the diagnosis of *Helicobacter pylori* gastritis based on endoscopic images; and Chen et. al [15] constructed a deep CNN based method for thorax disease diagnosis through chest x-ray images.

Quinn et. al [19] also evaluated the performance of deep CNN on three different microscopic tasks: diagnosis of malaria in thick blood smears, tuberculosis in sputum samples, and intestinal parasite egg in stool samples. The authors also explored methods to maximize the performance of the deep CNN they've constructed. In their methodology, the images were downsampled and then split up into overlapping patches, with the downsampling factor and patch size determined by the type of pathogen to be recognised in each case. This is because each image for each case have several objects (plasmodium for Malaria, Tuberculosis bacilli, and eggs of hookworm) of interest. Through this, the authors have obtained a significantly larger data set. Although the potential number of negative patches (i.e. with absence of any of these pathogens, though possible with other types of objects such as staining artifacts, blood cell or impurities) is disproportionately large compared to the number of positive patches since most of each image do not contain pathogen objects, two measures were done by the authors to make the training and testing sets more balanced. First, negative patches were randomly discarded so that there was at most 100 times the number of positive patches. Second, new positive patches were created by applying data augmenting techniques such as rotating, flipping and a combination of both, yielding 7 extra positive patches for each original one. Their method of generating more instances for the data set is especially useful for when the number of images procured is small. For the results of all cases, accuracy is very high with the AOC scores being greater than 0.99.

Cancer imaging is no exception to deep learning applications. In fact, there have been

several studies about deep learning applications for lung cancer, skin cancer, colorectal cancer, bladder cancer, prostate cancer, and breast cancer for the past couple of years. Pearce [20] investigated the application of deep learning networks to tumour classification specifically to correctly classify the location of mitoses in slide images. The author found that simple networks can deliver reasonable performance comparable with mid-range performers on the same data set. The simple network constructed in the study is a three-layer convolutional neural network with a binary classifier in the final layer trained on biopsy slide images. The author also explored ways in which model saturation can be resolved. The study suggests that model saturation can be resolved by a combination of limiting the number of parameters in the model, ensuring that training data is balanced between positive and negative instances, low learning rates, and iteratively biasing the input data towards examples that the model has incorrectly classified after previous training epochs (supervised learning). The resulting model performed reasonably well scoring a 72% accuracy.

Computer imaging methods in the past fail to consider the difference between natural images and medical images for training CNNs. Moreover, when examining medical images, a set of views must be fused in order to perform a proper reading. Geras et. al [21] proposed to use a multi-view deep CNN that handles a set of high-resolution medical images specifically a large-scale mammography-based breast cancer screening data set consisting of 886,000 images. The authors built a CNN that classifies an input image as BI-RADS 0 ("incomplete"), BI-RADS 1 ("normal"), or BI-RADS 2 ("benign finding"). The novelty of the authors' method here is that instead of heavily downscaling an original high-resolution image, they kept the input at 2600 x 2000 pixels. Heavily downscaling an image is a common method in object recognition and detection in order to improve the computational efficiency, both in terms of computation and memory, and also because no further improvement has been attributed with higher-resolution images. However, the authors argued that this kind of method works well with natural images where the objects of interest are usually presented in larger portions than other objects, and what matters most are their macro-structures such as shapes, colors, etc. On the other hand, medical images, where



there are fine details to consider, do not share these properties with natural images, so the downscaling of medical images are not desirable. The authors used aggressive convolution layers and pooling layers specifically convolution layers with strides larger than one in the first two convolutional layers and the first pooling layer with a larger stride than the other pooling layers. An experiment was conducted where the accuracies were evaluated with different decreases in resolution of the image. The results are that when the images were downscaled by  $\times 1/8$ , the peak accuracy was 74.3%. On the other hand, when the images are not downscaled, the peak accuracy was 78.7%.

In training and fine tuning CNNs, a large amount of labeled data is always required. However, for most medical image projects, acquiring such a database is very difficult. Sun et. al [22] developed a graph-based semi-supervised learning (SSL) scheme using deep CNN for breast cancer diagnosis. SSL is a technique in machine learning wherein labeled data and unlabeled data are employed in the training of the model. Here, the authors' proposed scheme only requires a small portion of labeled data (100 instances). This is potentially useful and powerful since the process of diagnosing and labeling hundreds or even thousands of data is costly. Consequently, their CNN model achieved an accuracy of 0.8243 and an area under the ROC curve of 0.8818.

According to Chougrad et. al [23], radiologists often find it difficult to classify mammography mass lesions. So, the authors constructed a Computer Assisted Diagnosis (CAD) system based on a deep CNN model that classifies breast mass lesions. The authors proposed the use of transfer learning with exponentially decaying learning rate per layer. The use of transfer learning was justified by the small size of the data set only comprising of 600 images, 300 images showing benign lesions and 300 showing malignant lesions. Then, when fine-tuning the CNN, the weights of the last convolutional layers need to be tweaked as much as possible, while the first ones remain nearly untouched. They propose to do this by the aforementioned per-layer exponentially decaying learning rate. The rationale for this was that the first layers of CNN learn generic features, while the last layers tend to be more specific to the data hence by fine-tuning the last convolutional layers, the model

learns more data-specific features. With the setup they've proposed, their model achieved an accuracy of 98.23% on an independent, test database.

Arajo et. al [24] constructed a CNN for a different task specifically the classification of hematoxylin and eosin stained breast biopsy images into four classes namely normal tissue, benign lesion, in situ carcinoma and invasive carcinoma, and in two classes, carcinoma and non-carcinoma. The architecture of the CNN was custom-tailored to retrieve information at nuclei level and overall tissue organization. This is performed by first processing several patches obtained from an input image with a patch-wise classifier, and then combining the classification results of all the image patches to obtain the final image-wise classification. In addition, the author evaluated the performance of two network architecture, a CNN and a CNN with the feed-forward neural network swapped with a Support Vector Machine (SVM) classifier. Accuracies of 77.8% for four class and 83.3% for carcinoma/non-carcinoma are achieved.

Similarly, Rakhlin et. al [25] had similar objectives to that of Arajo et. al which was to develop a deep CNN for the classification of hematoxylin and eosin stained breast biopsy images into four classes (normal tissue, benign lesion, in situ carcinoma and invasive carcinoma) and in two classes (carcinoma and non-carcinoma). They both had the same data set, 400 images of 4 classes. To remedy the problem of overfitting due to a small data set, the authors employed an approach known as deep convolutional feature representation by which deep CNNs, trained on large and general datasets like ImageNet, are used for unsupervised feature representation extraction. They also used Light GBM, a fast, distributed, high performance implementation of gradient boosted trees, for supervised classification. The results of their model outperformed that of [24] having an accuracy of 87.2% for the 4-class classification task and 93.8% accuracy for the 2-class classification.

In 2015, He et. al [17] proposed a residual learning framework to achieve deeper CNNs without dealing with vanishing and exploding gradients when training very deep CNNs. They constructed a CNN with shortcut connections that skip one or more stacked layers

through use of identity mapping where the outputs of shortcut connections are added to the outputs of the stacked layers. The resulting network is referred to as a residual neural network (RNN). The author’s constructed model, a 152-layer RNN, surpassed human-level performance on the ImageNet classification data set with an accuracy of 0.89.

Unfortunately, these studies do not discuss the implementation of their model into a widely-distributed system. However, Weng [26] was tasked to develop a CNN by iSono Health, a startup company, who developed the iSono app, an affordable, automated ultrasound imaging platform to assist women with monthly self-monitoring of breast cancer detection. The application comes with a device that allows user to self-scan through ultrasound. The resulting images are sent to the user’s physician who sends back a diagnosis regarding the images. A well-developed CNN could potentially remove the need for a physician, for the physician is not always readily-available. The objective of the CNN was to differentiate benign and malignant breast lesions. Furthermore, the author compared the performance of a CNN with respect to a fully-connected neural network (FCNN). The CNN model constructed achieved an accuracy of 73% while the FCNN only achieved 66%. In the author’s paper, there was no discussion regarding the specifics of incorporating the constructed model into the iSono app. Moreover, the CNN model achieved a relatively low accuracy compared to other models mentioned as the model was built from scratch, and the design was inspired by AlexNet.

Large-scale medical image databases with fully-annotated ROIs are scarce. There are only a handful of such databases that are publicly available such as CBIS-DDSM. It would also be unrealistic to require all existing databases to be fully annotated with ROIs. In [18], Shen developed an end-to-end training algorithm for whole-image breast cancer diagnosis based on mammograms. The authors achieved a trainable whole image classifier model from patch-wise classifier model greatly reducing reliance on data sets with lesion annotations as there are only a few databases that exist with such ROI annotations. This is done by training a patch-wise classifier model first then adding a new convolutional layer on top of the patch-wise classifier. The best constructed whole image classifier model achieved an

AUC score of 0.88 on DDSM (Digital Database for Screening Mammography). And, the model was trained as well on INbreast, a database with whole image annotations, which achieved an AUC score of 0.96.

### III. Theoretical Framework

#### A. Breast Cancer

Breast Cancer is a malignant tumor, a collection of cancer cells, arising from the cells of the breast. Breast cancer typically originates in the cells of the lobules, milk-producing glands, or the ducts, the passages that drain milk from the lobules to the nipple [1].

There are several types of breast cancer, but the type of breast cancer is classified by the specific cells in the breast that are afflicted by the disease. The most common types are ductal carcinoma in situ, invasive ductal carcinoma, and invasive lobular carcinoma. The terms "in situ" and "invasive" refer to the extent of the spread of the cancer. "In situ," also called as "non-invasive," means that the cancer has not spread, while "invasive," also called as "infiltrating," means that the cancer has spread to the surrounding breast tissue [27].

1. *Ductal carcinoma in situ*

Ductal carcinoma in situ (DCIS) is the most common type of non-invasive breast cancer [28]. DCIS is described by cancer cells that line the milk ducts of the breast. Since it is non-invasive, it is contained only in the milk ducts, but immediate treatment is still necessary, for it may eventually become invasive [29].

2. *Invasive ductal carcinoma*

Invasive ductal carcinoma (IDC) is the most common type of breast cancer accounting for about 80% of all breast cancers. IDC means that the cancer has spread to the surrounding breast tissue originating from the milk ducts. Over time, the cancer may spread to the lymph nodes and other parts of the body [30].

3. *Invasive lobular carcinoma*

Invasive lobular carcinoma (ILC) is the second most common type of breast cancer after IDC. ILC is characterized by cancer that originated from the lobules. Like IDC, the cancer cells have the potential to spread to the lymph nodes and other areas of the body [31].

As mentioned, the most common symptom of breast cancer is a new lump or mass. A painless, hard mass with irregular edges is evidently cancer, but, in other breast cancer cases, the mass can be tender, soft, rounded, and painful. Other possible symptoms include:

- swelling of all or part of a breast even if no distinct lump is felt
- skin irritation or dimpling
- breast or nipple pain
- nipple retraction
- redness, scaliness, or thickening of the nipple or breast skin
- nipple discharge

In invasive breast cancer, there are cases where, after spreading to the lymph nodes under the arm or around the collar bone, swelling may occur there, even before the original tumor in the breast becomes large enough to be felt [32].

The survival rates for breast cancer depends solely on the stage or the extent of the cancer. Generally, the survival rates are higher for women with earlier stage cancers. According to the National Cancer Institute's SEER database from people diagnosed with breast cancer from 2007 to 2013, the 5-year relative survival rate for women with stage 0 or stage I breast cancer is close to 100%; the 5-year relative survival rate for women with stage II breast cancer is approximately 93%; the 5-year survival rate for stage III breast cancers is about 72%; Lastly, the 5-year relative survival rate for women with stage IV breast cancers is about 22%. Fortunately, there are many widely available treatment options even for this stage of cancer [33].

Treatment depends on the biology and behavior of breast cancer. Treatment options and recommendations are very personalized and depend on several factors [34]:

- the stage of the tumor

- the patient's age, general health, menopausal status, and preferences
- the presence of known mutations in inherited breast cancer genes

Cancer care teams comprising of several health care professionals such as surgeons, radiologists, oncologists, physician assistants, oncology nurses, social workers, pharmacists, counselors, nutritionists, and others work together to create a patient's overall treatment plan which is a summary of planned cancer treatment. Some of the most common treatment options include [34]:

1. Surgery

Surgery is the removal of the tumor and some surrounding healthy tissue for good measure. There are two type of surgery namely lumpectomy and mastectomy. Lumpectomy is the removal of the tumor and a small, cancer-free area of healthy tissue around the tumor. Often, there could be several follow-up treatments after surgery such as radiation therapy. On the other hand, mastectomy is the surgical removal of the entire breast.

2. Radiation therapy

Radiation therapy is the use of high-energy x-rays of other particles to destroy cancer cells. Some types include external-beam radiation therapy, intra-operative radiation therapy, and brachytherapy. External-beam radiation therapy is radiation given to the affected area from a machine outside the body. Intra-operative radiation therapy is given using a probe in the operating room. Lastly, brachytherapy is done by placing radioactive sources into the tumor.

Radiation therapy is usually given over a set period of time indicated by the regimen given by the radiation oncologist. However, this kind of treatment may cause side effects such as fatigue, swelling of the breast, redness and/or skin discoloration, and pain/burning in the skin where the radiation was directed.

3. Chemotherapy

Chemotherapy is the use of drugs to destroy cancer cells usually by inhibiting the can-

cer cells' ability to grow and spread. The common ways by which chemotherapy drugs are applied include an intravenous tube placed into a vein using a needle, an injection under the skin or into a muscle, or a pill or capsule that is swallowed. Chemotherapy may also be given before surgery to shrink larger tumors to make surgery easier, and after surgery to minimize the risk of recurrence. Common drugs prescribed include capecitabine, carboplatin, cisplatin, cyclophosphamide, docetaxel, doxorubicin, pegylated liposomal doxorubicin, epirubicin, fluorouracil, gemcitabine, methotrexate, paclitaxel, protein-bound paclitaxel, vinorelbine, eribulin, ixabepilone, and others.

Like radiation therapy, chemotherapy is given over a set period of time for a number of cycles prescribed by a medical oncologist. Chemotherapy has its side effects as well, but it depends on the individual, the drugs used, and the schedule and dose used. These side effects include fatigue, risk of infection, nausea and vomiting, hair loss, loss of appetite, and diarrhea.

#### 4. Hormonal therapy

Hormonal therapy, also referred to as endocrine therapy, is an effective treatment only for tumors that test positive for either estrogen or progesterone receptors. Tumors with estrogen or progesterone receptors use these hormones to fuel their growth. Given this, hormonal therapy blocks the hormones to eliminate the cancer and prevent recurrences. Hormonal therapy, similar to chemotherapy, may be given before surgery to shrink a tumor and after a surgery to reduce the risk of recurrence. This kind of treatment is applied through a drug called tamoxifen or by aromatase inhibitors. Tamoxifen blocks estrogen from binding to breast cancer cells, effective for reducing the risk of recurrence. Aromatase inhibitors decrease the amount of estrogen made in tissues other than the ovaries in postmenopausal women by blocking the aromatase enzyme.

#### 5. Targeted therapy

Targeted therapy targets the resources (proteins, specific genes, tissue environment) that fuel the survival and growth of cancer. This type of treatment cuts off the



growth and spread of cancer cells as well as limiting damage to healthy cells. Some of the drugs prescribed for this kind of treatment include trastuzumab, pertuzumab, ado-trastuzumab emtansine, and neratinib.

With a multitude of treatment options widely available, it is still more effective to treat breast cancer at its earliest stage. In order to detect breast cancer at such an early stage where little to no symptoms are observable, screening examinations are given routinely to seemingly healthy individuals. The most common breast cancer screening examinations include [35]:

1. Clinical breast exam

Clinical breast exam is a physical examination of the breast conducted by a doctor or other medical professionals. The doctor meticulously checks the breasts and underarm area for lumps or changes in size or shape. However, this is not as effective and reliable as the other methods of screening breast cancer.

2. Mammography

Mammography is a low-dose x-ray exam. The x-ray images produced by this exam are called mammograms. During mammography, a radiologist will position the patient's breast in the mammography unit. The breast is placed on a special platform and compressed with a paddle. Then, the breast will be compressed gradually while the mammography unit takes the image producing a top-to-bottom view of the breast. The side view of the breast is also taken.

3. Breast magnetic resonance imaging

Breast magnetic resonance imaging (MRI) involves the use of a powerful magnetic field, radio frequency pulses, and a computer to produce detailed images of the insides of the breasts. MRI is not meant to replace mammography. Rather, MRI is used in conjunction with mammography and ultrasound to produce more accurate and detailed readings since it may identify abnormalities not visible with mammography or ultrasound.

During MRI, the patient will lie face down on a platform with openings meant for the breasts. A nurse or technologist will insert an intravenous catheter into a vein in the patient's hand or arm. The platform will be moved into the magnet of the MRI unit. Then, images will be taken while the patient remains still. Lastly, the contrast material is injected into the intravenous line, and additional images are taken.

#### 4. Breast ultrasound

Breast ultrasound uses sound waves to produce images of the inside of the breast. Like MRI, breast ultrasound is not a suitable replacement for mammography, but is more effective when used in conjunction with mammography and MRI. This is because breast ultrasound can cover areas of the breast that mammography can not. Also, breast ultrasound can identify whether a breast lump is a solid mass or a fluid-filled cyst.

In breast ultrasound, the patient will lie down on the examining table. A clear water-based gel is applied on the breast. The transducer will be placed firmly against the skin, sweeping over the breast, then the image is produced.

When the screening examination results show potential breast cancer, a diagnostic examination is conducted to determine the presence of breast cancer specifically a breast biopsy. A breast biopsy is an examination that removes tissue or fluid from the area of interest. The removed tissue is then carefully examined by a pathologist for abnormal or cancerous cells. Through a biopsy, an assessment report is produced whether or not the patient is positive for breast cancer [36].

In this paper, screening mammography is the topic of interest for an application of deep learning. For this purpose, screening mammography will be discussed in detail in the next section.

## B. Screening Mammography

A screening mammogram typically involves two radiographic images taken from two views of each breast. Craniocaudal (CC) images are radiographic images taken from the top of the breast, while mediolateral oblique (MLO) images are taken from the side. Mammographic results are standardized to be reported using final assessment categories of the Breast Imaging Reporting and Data System (BI-RADS) in order to create a uniform system with a recommendation for the course of treatment with each category [37].

In mammogram analysis, mammograms are positioned on a viewbox in a mirror-image fashion with both the MLO and CC views mounted back to back as shown in figure 1. There are two steps involved in mammogram analysis. The first step involves detecting a difference in structural aspect of the left and right breasts. The second step deals with analyzing this aspect to detect more or less physiological abnormalities, or to classify the image as suspicious of malignancy. There are three major types of abnormalities that may be found specifically asymmetrical densities, masses and architectural distortions, and calcifications [38]. For the purpose of this paper, masses and calcifications will only be discussed.

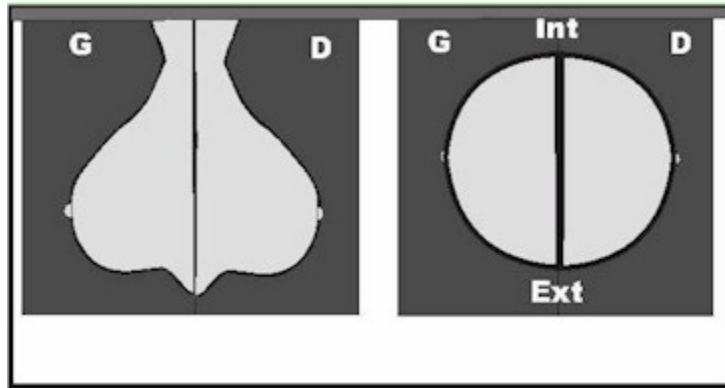


Figure 1: A back to back position of the left and right breasts of both the MLO and CC views, Mammo

Masses are three-dimensional lesions in the sense that they're seen on multiple views. The location of a mass may be determined through the analysis of multiple views where observable masses are suspicious of malignancy. Size does not determine the malignancy of

a mass, except if on successive views, the size of the mass regularly increases. Masses can be distinguished in five different types presented in figure 2: round, oval, lobular, irregular, and architectural distortion [39].



Figure 2: The five different types of mammogram masses, Mammo

Mass margins must also be detailed during analysis specifically the ROI they cover. As such, the five different margin types include circumscribed (well-defined and sharply demarcated), microlobulated (small circled line the edges), obscured, indistinct or ill-defined, and spiculated as seen in figure 3. The obscured margins are due to adjacent normal tissue overlapping. Indistinct or spiculated margins illustrate the invasion of the malignant tumour into surrounding healthy tissue [39].



Figure 3: The five different types of mammogram mass margins, Mammo

Generally, round, oval, or lobulated masses with circumscribed limits are benign tumors with only little known cases not following this rule [40]. Round masses with obscured limits are difficult to analyze often mimicking a cancer mass with ill-defined borders. In light of this, further breast examinations are performed. Masses with lobulated or microlobulated limits are suspicious of malignancy where the more lobulated the limits are, the greater the risk of malignancy. Masses with indistinct limits are also suspicious of malignancy [41]. Masses observed as architectural distortions may also be a sign of malignancy, but they may also be a result of a surgical scar or unrelated diseases. Consequently, further breast examination must be performed. Masses with spiculated borders are highly suspicious of

cancer as spicules illustrate invasion of the tumour into surrounding tissue [42].

Breast calcifications are small calcium deposits in women’s breast tissue [43]. Calcification images should be detailed according to size, shape, number of calcifications, and distribution. Generally, larger calcifications (macrocalcifications) with regular or oval shapes are benign, while smaller, irregular calcifications (microcalcifications) are potentially malignant. Moreover, in analysing the size of calcifications, the sizes of microcalcification fall between 0.2 - 0.5 mm, while macrocalcification sizes are 2.0 mm or larger [44].

When detailing the shape of calcifications, round, oval calcifications with uniform shape and size are suggestively benign, while irregular calcifications are potentially malignant. Furthermore, M. Le Gal proposed five types of microcalcifications according to shape with corresponding degrees of malignancy as detailed in figure 4: type I (tea-cup, annular, clear centres) with 0% malignancy potential, type II (regularly punctiform) with 39% malignancy potential, type III (dusty, salt particles) with 39% malignancy potential, type IV (irregularly punctiform) with 59% malignancy potential, and type V (vermicular punctiform) with 96% malignancy potential [44].

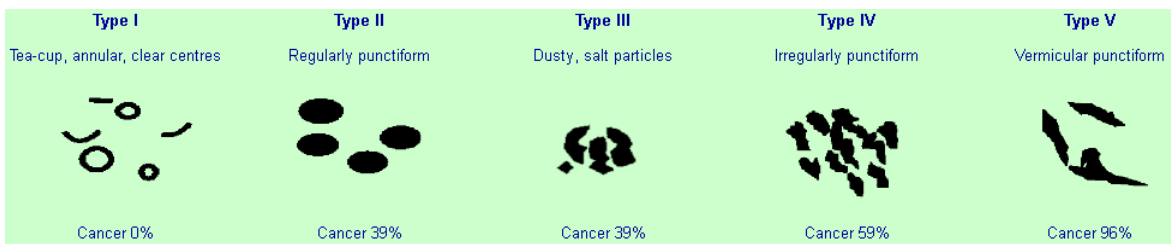


Figure 4: The five different types of microcalcifications based on shape, Mammo

Lastly, when analysing the number of calcifications, any number above four up to six microcalcifications is indicative of malignancy [44].

After mammographic analysis and evaluation, the results are classified according to one of the following BI-RADS categories [37]:

1. Category 0: Additional imaging evaluation and/or comparison to prior mammograms

is needed

The findings in the mammogram require additional imaging examination. Additional imaging examinations include ultrasound, MRI, special mammogram views, spot compression, and magnified views. Moreover, the mammogram of interest may be compared to past mammograms to identify changes over a period of time.

2. Category 1: Negative

The mammogram under study contain no significant abnormalities indicative of malignancy. The breasts appear healthy.

3. Category 2: Benign Findings

Like category 1, the mammogram of interest appear to contain no malignant abnormalities, but benign findings are present. Typical findings include benign-appearing macrocalcifications, oil cyst, or a lipoma.

4. Category 3: Probably Benign Findings; Short-Interval Follow-up Suggested

This is a mammogram that is usually benign, but further exploration should be performed to generate more stable readings.

5. Category 4: Suspicious Abnormality; Biopsy Should be Considered

The mammogram analyzed contains possible malignant abnormalities but not obviously malignant mammographically. A breast biopsy is recommended to verify malignancy.

6. Category 5: Highly Suggestive of Malignancy; Appropriate Action Should be Taken

The findings present in the mammogram are highly probable (> 95%) of being malignant. Typical findings include spiculated mass or malignant-appearing microcalcifications. As such, breast biopsy is also recommended.

7. Category 6: Known Biopsy-Proven Malignancy; Appropriate Action Should be Taken

The findings on the mammogram have already been verified as malignant by a previous breast biopsy. This category is for mammograms that have cancer under study to see how well the cancer is responding to a particular treatment.

## C. Machine Learning

Machine learning is the science behind the ability of computers to act and learn concepts without being explicitly programmed, and improve learning over time through data and observations [45]. To date, there are so many machine learning algorithms each with different uses for specific functions. Machine learning algorithms can be categorized into four types based on their purpose [46]:

### 1. Supervised Learning

In supervised learning problems, a data set is procured containing training examples or instances with associated correct labels or predictions. For example, the problem is to have the computer learn to classify handwritten digits. In a supervised learning approach, a data set must be acquired with thousands of pictures of handwritten digits along with labels identifying the correct number illustrated by image. The implemented algorithm would attempt to learn and acquire features from the images with respect to the number they represent in order to build a model to classify said images. The model would undergo several parameter tweakings and adjustments to achieve an optimal model. This process is referred to as training. Then, the model would be put to the test to classify handwritten digits on an unlabeled data set [47]. Algorithms under supervised learning include Nearest Neighbor, Naive Bayes, Decision Trees, Linear Regression, Support Vector Machines (SVM), and Neural Networks.

### 2. Unsupervised Learning

In contrast to supervised learning, unsupervised learning involves training the computer with unlabeled data. This type of machine learning is typically used for pattern detection and descriptive modeling. Since there are no labels for which an algorithm can model relationships, unsupervised learning algorithms attempt to utilize techniques on the data to mine for rules, detect patterns, and summarize and group the data. Some common algorithms include k-means clustering and Association Rules.

### 3. Semi-supervised Learning

Semi-supervised learning involves a combination of the first two types. In order for

supervised learning to be at its most effective, a large labeled data set is required. However, in practical situations, the cost of labeling is high, and, in some fields, a large fully-labeled data set lacks. So, in the absence of correct labels in the majority of the observations but present in few, semi-supervised learning algorithms may be implemented.

Furthermore, there are algorithms centered on a specific technique of machine learning called deep learning (DL). The main advantage of DL algorithms over ML algorithms is that DL algorithms can generate new features from existing ones in the training data set, while in ML algorithms, the features must be accurately identified which can be costly. Therefore, DL algorithms save more time when dealing with large, big complex data [49].

Artificial neural networks (ANN), inspired by the biological structure of the human brain, are machine learning algorithms applied on the computer to perform specific tasks such as clustering, classification, pattern recognition, etc. A typical ANN contains hundreds of interconnected single units, artificial neurons, connected with coefficients (weights), which constitute the neural structure and are grouped and separated in layers. The behavior of an ANN depends on several factors such as the activation function of a neuron, the weights of each neuron, biases of each layer, the learning rule of the ANN, the architecture itself, etc. The output of a single neuron is determined by the weighted sum of the inputs passed through the activation function [50].

During training, the training set is passed through the network, and the output obtained is compared with the actual value. The difference between the predicted output and the actual value is referred to as the error. The objective of training the network is to achieve a model where the parameters are optimized in such a way that the error in predictions is minimized specifically the error converges to a local minimum [50].



## D. Convolutional Neural Networks

Convolutional Neural Network (CNN) is a common DL algorithm and a special case of ANNs. Historically, CNNs are mainly used for image recognition tasks specifically object detection and classification, due to their ability to learn higher-order features. A CNN is primarily composed of one or more convolutional and pooling (or subsampling) layers and then followed by one or more fully-connected layers. Figure 5 illustrates how a standard CNN architecture handles a vehicle image classification task. An image is input into the network where it undergoes several stages of convolution and pooling. The features extracted from the process are fed into the fully-connected neural network. Finally, the last fully-connected neural network outputs the predicted classification [51].

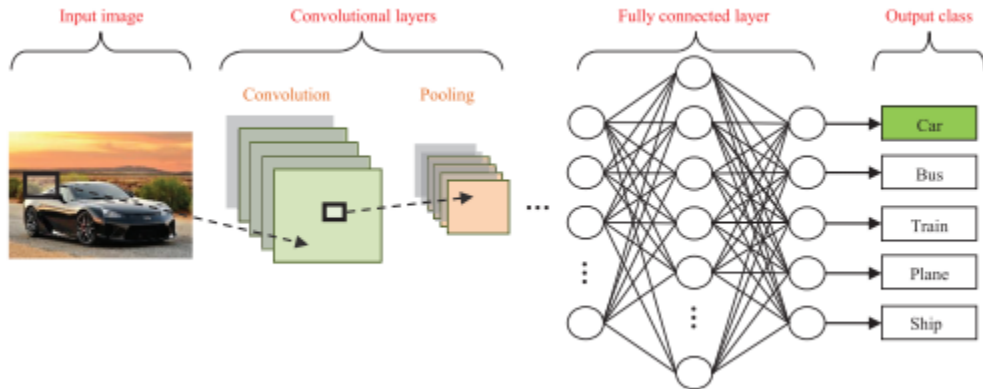


Figure 5: An intuitive overview of a standard CNN, Mammo

In the convolutional layer, features are extracted, and the feature representations of the input image is obtained. The convolutional layer's parameters involves a set of learnable filters typically small in dimension but extends through the full depth of the input volume (i.e. a filter for the first layer of size  $5 \times 5 \times 3$ , with  $5 \times 5$  height and width, and depth 3 for color channels). In the processing of the input volume, the filter slides (convolves) across the width and height of the input volume, while dot products are computed from the filter entries and the input. Typically, the first convolutional layers learn low-level features like edges or blotches of some color, while the latter layers learn higher level features [52]. Figure 6 illustrates how convolution works.

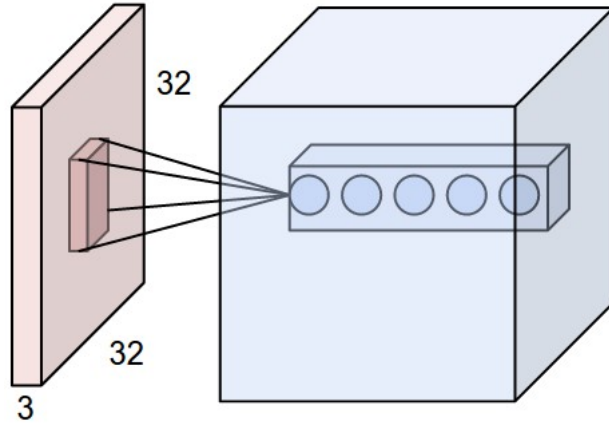


Figure 6: An illustration of an input volume (i.e. 32x32x3 input image) at the left and the first convolutional layer (a volume of neurons) to the right; image, Mammo

The pooling layer, commonly placed in succession with several convolutional layers, serves to progressively reduce the spatial size of the representation in order to reduce the amount of parameters and computational costs in the network. There are two types of pooling namely average pooling and max pooling. Figure 7 illustrates the difference between the two types [51].

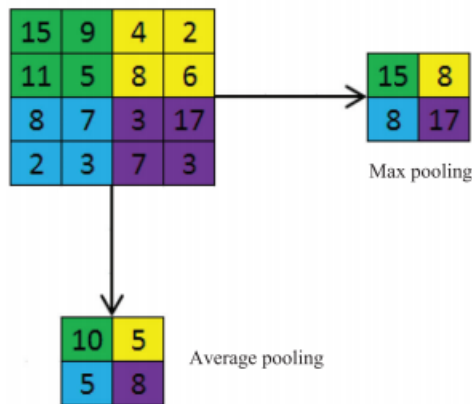


Figure 7: Average pooling vs max pooling, Mammo

In figure 7, a common pooling layer form is used where the filter size is 2x2 applied with a stride of 2. The input image here has a size of 4x4. After pooling is applied, max pooling outputs the maximum value of each 2x2 region, while average pooling outputs the average of each region.

Lastly, the fully-connected layers interpret the feature representations extracted from the convolutional and pooling layers in order to arrive at a conclusion. The last layer here outputs the predicted class label.

## E. Residual Neural Networks

According to He et. al [17], the motivation behind Residual Neural Networks (ResNet) is a property of CNNs: as more layers are stacked (as the CNN goes deeper), the levels of features integrated by the network are enriched which would entail better model generalization. However, as a CNN goes deeper, the network becomes more difficult to train due to the problem of vanishing and exploding gradients. Moreover, when deeper networks are able to start converging, a degradation problem occurs where, due to the increase in network depth, the accuracy becomes saturated and then degrades rapidly resulting to higher training error.

In their study [17], the authors attempt to address the degradation problem by introducing a deep residual learning framework. In their network, instead of having some stacked layers fit a desired underlying mapping, they let the stacked layers fit a residual mapping. The authors have hypothesized that it is easier to optimize the residual mapping than to optimize the underlying, original mapping. The construction of such residual mapping can be realized by CNNs with "shortcut connections." Shortcut connections are those skipping one or more layers through the use of identity mapping, where the outputs are added to the outputs of the stacked layers as seen in figure 8.

The authors have provided comprehensive evidence that their proposed residual learning framework is effective as they've obtained excellent results on the ImageNet classification data set with their 152-layer RNN where they obtained a 3.57% top-5 error, and won 1st place in the ILSVRC 2015.

As stated, the network architecture used is a pretrained ResNet-50 which is a ResNet with 50 parameter layers.

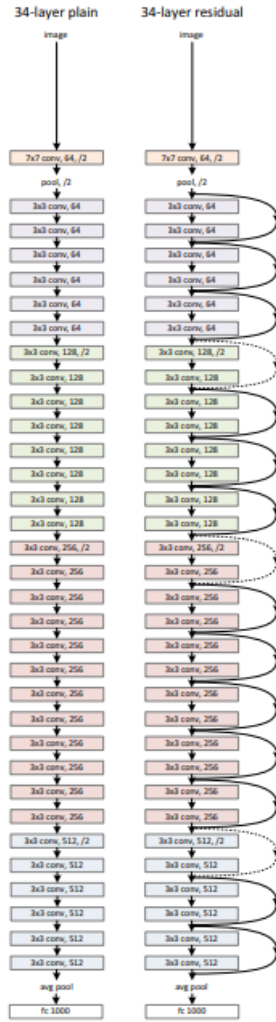


Figure 8: Left: a plain CNN with 34 parameter layers. Right: a RNN with 34 parameter layers. The dotted shortcuts increase dimensions, Mammo

## IV. Design and Implementation

### A. System Overview

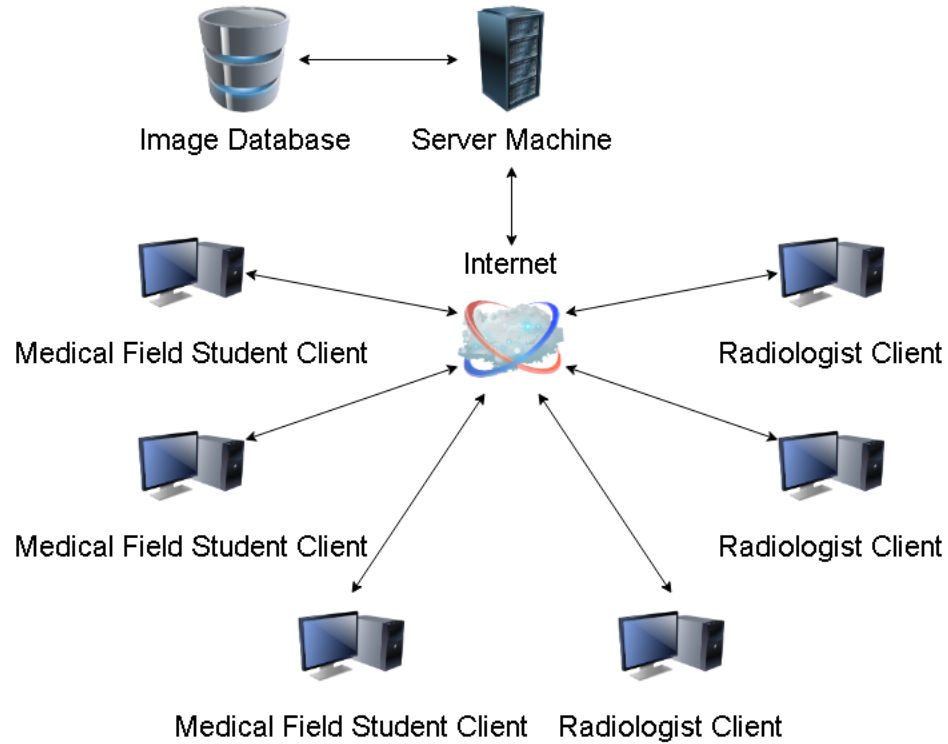


Figure 9: System Overview, Mammo

The Decision Support System includes a typical client-server architecture. The client sends requests to the server. The server processes the client's request. There are two types of clients namely the Medical Field Student Client (MFSC) and the Radiologist Client (RC). The MFSC and the RC can send mammogram images through desktop computers for evaluation to the server by which the server, in turn, returns the results. The MFSC may also request for an exam, and the server redirects the MFSC to the exam proper.

## B. Use Case Diagram

Figure 10 shows the general functionalities of each user. Both the Medical Field Student and Radiologist have the functionality of uploading a mammogram image for the system to classify whether benign or malignant, and retrieving the classification result. The Medical Field Student also has the option to take an exam on screening mammograms.

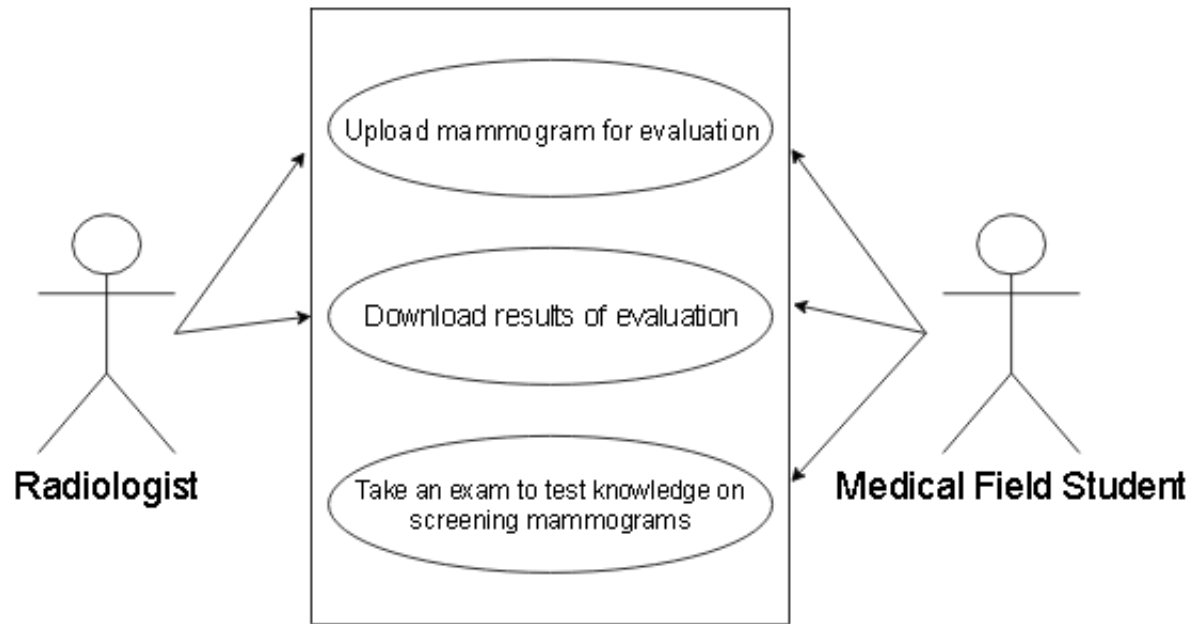


Figure 10: Use Case Diagram, Mammo

## C. Activity Diagram

### 1. Radiologist

The radiologist can also send a mammogram image for evaluation using the web application.

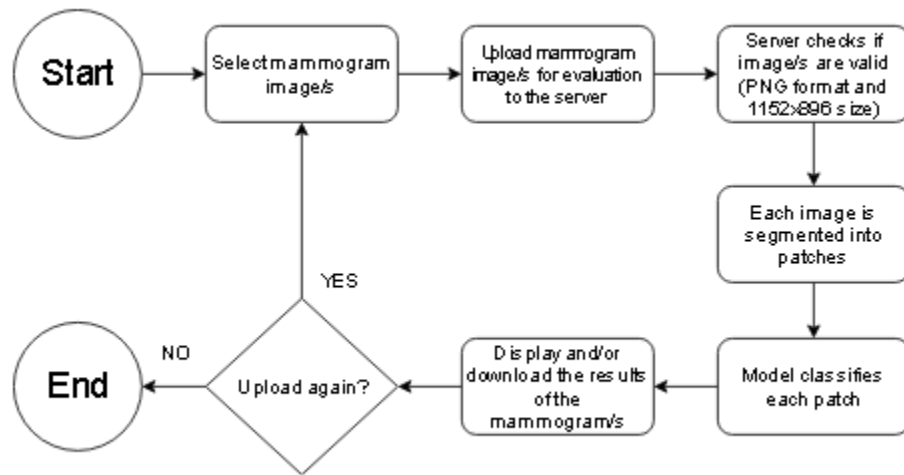


Figure 11: Radiologist Activity Diagram, Mammo

## 2. Medical Field Student

The medical field student can send a mammogram image for evaluation using the web application. The medical field student may also request for an exam on screening mammograms. The results of the exam are given immediately after.

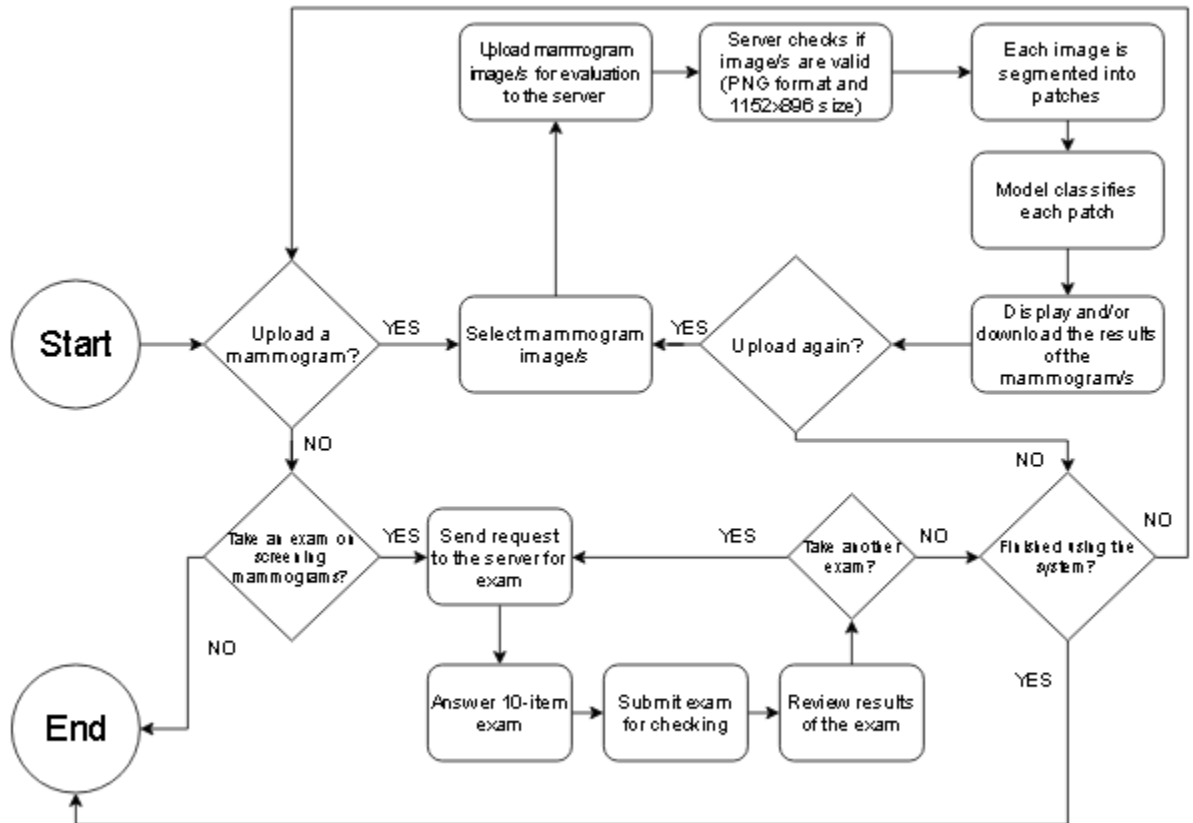


Figure 12: Medical Field Student Activity Diagram, Mammo



## D. Technical Architecture

The recommended requirements for the server machine include:

- 2 GHz CPU rate or higher
- Graphics Processing Unit (GPU) specifically a NVIDIA Graphics Card with 3.0 compute capability or higher
- 8 GB RAM or higher
- Up to 2 GB of free disk space

The client side must have any of the following compatible up-to-date web browsers:

- Google Chrome
- Mozilla Firefox
- Safari

## E. Data Set

The data set used is the CBIS-DDSM (Curated Breast Imaging Subset of the Digital Database for Screening Mammography) which is an updated and standardized version of the DDSM [53]. The data set contains 2583 cropped mammogram ROI images in DICOM format of 1249 patients labeled as benign or malignant cases with verified pathology information. It includes the CC and MLO views for most of the screened breasts. The data set is publicly available through TCIA (The Cancer Imaging Archive) which is an online service that hosts a large archive of medical images of cancer accessible for public download [54].

## V. Results

### Training the Convolutional Neural Network

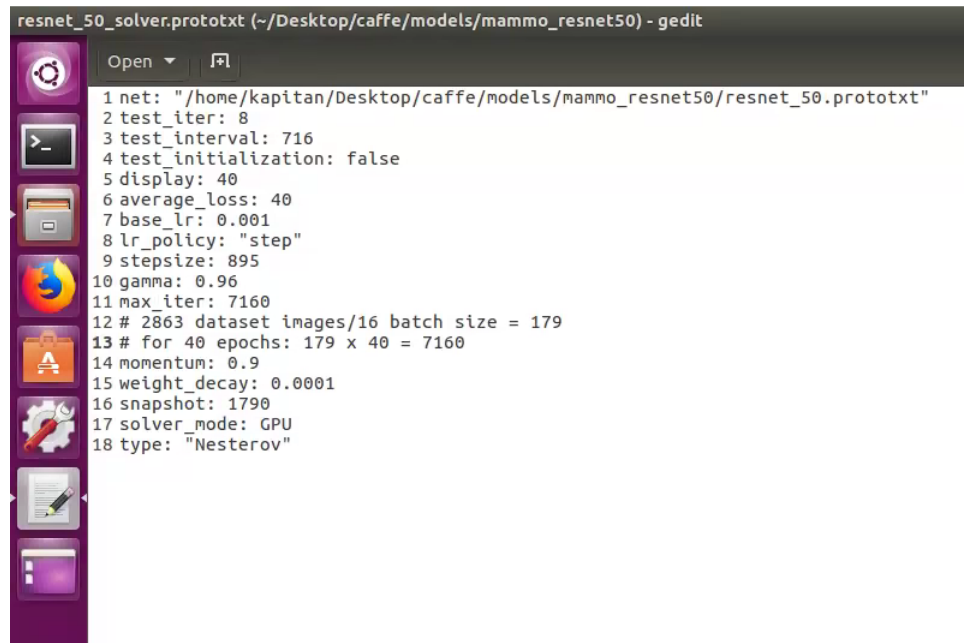
The dataset, CBIS-DDSM (Curated Breast Imaging Subset of the Digital Database for Screening Mammography), was downloaded from TCIA (The Cancer Imaging Archive). The dataset downloaded was similar to that in Shen’s study [18]. But, instead of the whole-image dataset version of CBIS-DDSM, the dataset downloaded was the cropped region of interest (ROI) version.

#### 1. Preparing the Dataset

Preparing the dataset was no easy task, for the dataset was erroneous. For context, there are two file locations in the given CSV metadata file for the whole dataset: one for the cropped ROI, and the other for the ROI mask. There are several occurrences where the file location of the cropped ROI was mislabeled and would point to the ROI mask instead. Since there are several occurrences in the dataset’s metadata where file locations are erroneous, there is no way of confirming if the other information are erroneous as well. This poses a potential problem where the ROIs are misclassified. It is this for this reason that the erroneous instances are kept track of.

#### 2. Setting the Hyperparameters

After setting the proper number of epochs, learning rate, batch size, and the optimizer namely Nesterov Gradient Descent seen in figure 13 as well as utilizing a pre-trained ResNet-50 model, and having the model undergo initial training, the model looked promising as it slowly converges shown in figure 14 for the first few iterations and figure 15 for the last remaining iterations. However, after having the model tested, it only attained an accuracy of 33% as it correctly labeled 243 images from a test set of 704. A possible cause for such a low accuracy is the lack of data augmentation techniques applied on the dataset used. Another cause is the imbalanced number of instances per category specifically there are 681 ROI images for benign mass, 637 ROI images for malignant mass, 1,002 ROI images for benign calcification, and 544 ROI images for malignant calcification.



```
1 net: "/home/kapitan/Desktop/caffe/models/mammo_resnet50/resnet_50.prototxt"
2 test_iter: 8
3 test_interval: 716
4 test_initialization: false
5 display: 40
6 average_loss: 40
7 base_lr: 0.001
8 lr_policy: "step"
9 stepsize: 895
10 gamma: 0.96
11 max_iter: 7160
12 # 2863 dataset images/16 batch size = 179
13 # for 40 epochs: 179 x 40 = 7160
14 momentum: 0.9
15 weight_decay: 0.0001
16 snapshot: 1790
17 solver_mode: GPU
18 type: "Nesterov"
```

Figure 13: ResNet-50’s hyperparameters, Mammo

### 3. Improving the Accuracy

To address the low accuracy, the dataset was balanced having the number of images under each category set at a maximum of 544 ROI images. This was done by removing the extra erroneous instances from each category, and then randomly removing other ones until it reached 544 images except images under malignant calcification. Then the model was trained again with the same hyperparameters except for the number of epochs which was increased from 40 to 50, and the use of the balanced dataset. Figures 16 and 17 show the model’s convergence at the first few iterations, and at the last remaining iterations. The accuracy improved up to 43% with the model correctly classifying 305 images out of 704 as seen in figure 18.

### General User Functionalities

The home page of the system is seen in figure 19. A basic description of what the site is for is present here along with the features it has to offer. Also, a navbar that provides the user access to the site’s features is present at all pages of the site.

```

model.log [Read-Only] (-/Desktop/caffe/models/mammo_resnet50) - gedit
8366 I0522 17:14:12.945467 27530 net.cpp:744] Ignoring source layer score
8367 I0522 17:14:12.945471 27530 net.cpp:744] Ignoring source layer loss
8368 I0522 17:14:12.945962 27530 solver.cpp:57] Solver scaffolding done.
8369 I0522 17:14:12.955227 27530 caffe.cpp:239] Starting Optimization
8370 I0522 17:14:12.955242 27530 solver.cpp:293] Solving ResNet-50
8371 I0522 17:14:12.955245 27530 solver.cpp:294] Learning Rate Policy: step
8372 I0522 17:14:12.955242 27530 solver.cpp:239] Iteration 0 (0 iter/s, 0.427552s/40 iters), loss = 1.66514
8373 I0522 17:14:13.382921 27530 solver.cpp:258] Train net output #0: prob = 1.66514 (* 1 = 1.66514 loss)
8374 I0522 17:14:13.382954 27530 sgd_solver.cpp:105] Iteration 0, lr = 0.001
8375 I0522 17:14:27.065359 27530 solver.cpp:239] Iteration 40 (2.92348 iter/s, 13.6823s/40 iters), loss = 1.48229
8376 I0522 17:14:27.065415 27530 solver.cpp:258] Train net output #0: prob = 1.44905 (* 1 = 1.44905 loss)
8377 I0522 17:14:27.065424 27530 sgd_solver.cpp:105] Iteration 40, lr = 0.001
8378 I0522 17:14:40.729054 27530 solver.cpp:239] Iteration 80 (2.9275 iter/s, 13.6635s/40 iters), loss = 1.5043
8379 I0522 17:14:40.729118 27530 solver.cpp:258] Train net output #0: prob = 1.7643 (* 1 = 1.7643 loss)
8380 I0522 17:14:40.729141 27530 sgd_solver.cpp:105] Iteration 80, lr = 0.001
8381 I0522 17:14:48.648066 27530 data_layer.cpp:73] Restarting data prefetching from start.
8382 I0522 17:14:54.485440 27530 solver.cpp:239] Iteration 120 (2.90778 iter/s, 13.7562s/40 iters), loss = 1.45234
8383 I0522 17:14:54.485492 27530 solver.cpp:258] Train net output #0: prob = 1.65897 (* 1 = 1.65897 loss)
8384 I0522 17:14:54.485500 27530 sgd_solver.cpp:105] Iteration 120, lr = 0.001
8385 I0522 17:15:08.214148 27530 solver.cpp:239] Iteration 160 (2.91363 iter/s, 13.7286s/40 iters), loss = 1.36864
8386 I0522 17:15:08.214210 27530 solver.cpp:258] Train net output #0: prob = 1.46695 (* 1 = 1.46695 loss)
8387 I0522 17:15:08.214234 27530 sgd_solver.cpp:105] Iteration 160, lr = 0.001
8388 I0522 17:15:21.878106 27530 solver.cpp:239] Iteration 200 (2.92744 iter/s, 13.6638s/40 iters), loss = 1.36951
8389 I0522 17:15:21.878293 27530 solver.cpp:258] Train net output #0: prob = 1.10346 (* 1 = 1.10346 loss)
8390 I0522 17:15:21.878302 27530 sgd_solver.cpp:105] Iteration 200, lr = 0.001
8391 I0522 17:15:25.349967 27530 data_layer.cpp:73] Restarting data prefetching from start.
8392 I0522 17:15:35.605042 27530 solver.cpp:239] Iteration 240 (2.91404 iter/s, 13.7267s/40 iters), loss = 1.30066
8393 I0522 17:15:35.605104 27530 solver.cpp:258] Train net output #0: prob = 1.21625 (* 1 = 1.21625 loss)
8394 I0522 17:15:35.605128 27530 sgd_solver.cpp:105] Iteration 240, lr = 0.001
8395 I0522 17:15:49.272727 27530 solver.cpp:239] Iteration 280 (2.92664 iter/s, 13.6675s/40 iters), loss = 1.34095
8396 I0522 17:15:49.272779 27530 solver.cpp:258] Train net output #0: prob = 1.2469 (* 1 = 1.2469 loss)
8397 I0522 17:15:49.272789 27530 sgd_solver.cpp:105] Iteration 280, lr = 0.001
8398 I0522 17:16:02.318789 27530 data_layer.cpp:73] Restarting data prefetching from start.
8399 I0522 17:16:02.993812 27530 solver.cpp:239] Iteration 320 (2.91542 iter/s, 13.7202s/40 iters), loss = 1.32628
8400 I0522 17:16:02.993077 27530 solver.cpp:258] Train net output #0: prob = 1.27098 (* 1 = 1.27098 loss)
8401 I0522 17:16:02.993099 27530 sgd_solver.cpp:105] Iteration 320, lr = 0.001

```

Figure 14: The log file of the model showing diminishing loss on the first few iterations as indicated by the red arrows, Mammo

## A. Mammogram Analysis

### 1. Uploading Mammograms

Figure 20 shows that a user has fully uploaded his chosen mammograms. A user may do this through the "Add Images" button at the top of the left sidebar or by just dragging and dropping files from his/her directory.

### 2. Editing and Deleting Mammograms

The mammograms are uploaded with unregistered metadata that is up for the user to edit. A user may edit the mammograms by clicking the edit icon present at the upper right of the mammogram cards, or he may simple click on the picture of the mammogram as shown figure 21. Also, a user may delete the mammogram by pressing the delete icon next to the edit. Moreover, a user may delete all the mammograms uploaded by the "Delete All" button on the sidebar. Also, a user must specify the region of interest to be analyzed by the CNN module. A user may leave the metadata entries blank, but he must choose a

```

model.log [Read-Only] [~/Desktop/caffe/models/mammo_resnet50] - gedit
9150 10522 17:52:43.595299 27530 solver.cpp:258] Train net output #0: prob = 0.00127987 (* 1 = 0.00127987 loss)
9157 10522 17:52:43.595325 27530 sgd_solver.cpp:105] Iteration 6760, lr = 0.000751447
9158 10522 17:52:44.992372 27530 data_layer.cpp:73] Restarting data prefetching from start.
9159 10522 17:52:57.291405 27530 solver.cpp:239] Iteration 6800 (2.92052 iter/s, 13.6962s/40 iters), loss = 0.050524
9160 10522 17:52:57.291458 27530 solver.cpp:258] Train net output #0: prob = 0.00872535 (* 1 = 0.00872535 loss)
9161 10522 17:52:57.291466 27530 sgd_solver.cpp:105] Iteration 6800, lr = 0.000751447
9162 10522 17:53:11.120769 27530 solver.cpp:239] Iteration 6840 (2.8924 iter/s, 13.8294s/40 iters), loss = 0.0471909
9163 10522 17:53:11.120829 27530 solver.cpp:258] Train net output #0: prob = 0.064486 (* 1 = 0.064486 loss)
9164 10522 17:53:11.120839 27530 sgd_solver.cpp:105] Iteration 6840, lr = 0.000751447
9165 10522 17:53:22.012444 27530 data_layer.cpp:73] Restarting data prefetching from start.
9166 10522 17:53:25.046046 27530 solver.cpp:239] Iteration 6880 (2.87248 iter/s, 13.9253s/40 iters), loss = 0.0676628
9167 10522 17:53:25.046100 27530 solver.cpp:258] Train net output #0: prob = 0.0321162 (* 1 = 0.0321162 loss)
9168 10522 17:53:25.046108 27530 sgd_solver.cpp:105] Iteration 6880, lr = 0.000751447
9169 10522 17:53:38.613261 27530 solver.cpp:239] Iteration 6920 (2.94828 iter/s, 13.5672s/40 iters), loss = 0.0463074
9170 10522 17:53:38.613315 27530 solver.cpp:258] Train net output #0: prob = 0.225407 (* 1 = 0.225407 loss)
9171 10522 17:53:38.613323 27530 sgd_solver.cpp:105] Iteration 6920, lr = 0.000751447
9172 10522 17:53:52.170032 27530 solver.cpp:239] Iteration 6960 (2.95055 iter/s, 13.5568s/40 iters), loss = 0.0472365
9173 10522 17:53:52.170233 27530 solver.cpp:258] Train net output #0: prob = 0.0240614 (* 1 = 0.0240614 loss)
9174 10522 17:53:52.170258 27530 sgd_solver.cpp:105] Iteration 6960, lr = 0.000751447
9175 10522 17:53:58.633590 27530 data_layer.cpp:73] Restarting data prefetching from start.
9176 10522 17:54:05.732872 27530 solver.cpp:239] Iteration 7000 (2.94926 iter/s, 13.5627s/40 iters), loss = 0.0319566
9177 10522 17:54:05.732931 27530 solver.cpp:258] Train net output #0: prob = 0.023159 (* 1 = 0.023159 loss)
9178 10522 17:54:05.732939 27530 sgd_solver.cpp:105] Iteration 7000, lr = 0.000751447
9179 10522 17:54:19.287967 27530 solver.cpp:239] Iteration 7040 (2.95092 iter/s, 13.5551s/40 iters), loss = 0.047832
9180 10522 17:54:19.288022 27530 solver.cpp:258] Train net output #0: prob = 0.0209382 (* 1 = 0.0209382 loss)
9181 10522 17:54:19.288029 27530 sgd_solver.cpp:105] Iteration 7040, lr = 0.000751447
9182 10522 17:54:32.853919 27530 solver.cpp:239] Iteration 7080 (2.94856 iter/s, 13.566s/40 iters), loss = 0.0276609
9183 10522 17:54:32.854128 27530 solver.cpp:258] Train net output #0: prob = 0.000721315 (* 1 = 0.000721315 loss)
9184 10522 17:54:32.854153 27530 sgd_solver.cpp:105] Iteration 7080, lr = 0.000751447
9185 10522 17:54:34.913853 27530 data_layer.cpp:73] Restarting data prefetching from start.
9186 10522 17:54:46.428627 27530 solver.cpp:239] Iteration 7120 (2.94669 iter/s, 13.5746s/40 iters), loss = 0.0381517
9187 10522 17:54:46.428683 27530 solver.cpp:258] Train net output #0: prob = 0.0193162 (* 1 = 0.0193162 loss)
9188 10522 17:54:46.428690 27530 sgd_solver.cpp:105] Iteration 7120, lr = 0.000751447
9189 10522 17:54:59.662269 27530 solver.cpp:468] Snapshotting to binary proto file /home/kapitan/Desktop/caffe/models/mammo_resnet50/
resnet_50_solver_iter_7160.caffemodel
9190 10522 17:54:59.878772 27530 sgd_solver.cpp:273] Snapshotting solver state to binary proto file /home/kapitan/Desktop/caffe/models/
mammo_resnet50/resnet_50_solver_iter_7160.solverstate

```

Figure 15: The log file of the model showing diminishing loss on the last remaining iterations as indicated by the red arrows, Mammo

region of interest and save it.

### 3. Analyzing the Mammograms

After the user has entered the necessary patient data, he/she may then proceed to have the mammograms analyzed by the "Analyze Mammograms" on the sidebar (see figure 20). It may take a long time to process the mammograms. Figures 22 and 23 show the analysis made by the CNN module. A "Generate PDF" functionality is present for the user to save the results as PDF (see figure 24).

## B. Quiz

### 1. Taking the Quiz

A user or a student/trainee has the option to take a randomly-generated 10-item pop quiz regarding breast cancer, mammography, treatment, etc. (see figure 25).

```

mammo_resnet50_balanced_model.log [Read-Only] (-/Desktop/caffe/models/mammo_resnet50_balanced) - gedit
8364 I0526 18:18:37.293448 1208 net.cpp:744] Ignoring source layer last_relu
8365 I0526 18:18:37.293452 1208 net.cpp:744] Ignoring source layer global_pool
8366 I0526 18:18:37.293454 1208 net.cpp:744] Ignoring source layer score
8367 I0526 18:18:37.293457 1208 net.cpp:744] Ignoring source layer loss
8368 I0526 18:18:37.293998 1208 solver.cpp:57] Solver scaffolding done.
8369 I0526 18:18:37.302898 1208 caffe.cpp:239] Starting Optimization
8370 I0526 18:18:37.302914 1208 solver.cpp:293] Solving ResNet-50
8371 I0526 18:18:37.302917 1208 solver.cpp:294] Learning Rate Policy: step
8372 I0526 18:18:37.727984 1208 solver.cpp:239] Iteration 0 (6.66909e+08 iter/s, 0.425009s/40 iters), loss = 1.93327
1208 solver.cpp:258] Train net output #0: prob = 1.93327 (* 1 = 1.93327 loss)
8374 I0526 18:18:37.728070 1208 sgd_solver.cpp:105] Iteration 0, lr = 0.001
8375 I0526 18:18:51.415144 1208 solver.cpp:239] Iteration 40 (2.92242 iter/s, 13.6873s/40 iters), loss = 1.49934
1208 solver.cpp:258] Train net output #0: prob = 1.47312 (* 1 = 1.47312 loss)
8376 I0526 18:18:51.415196 1208 solver.cpp:258] Train net output #0: prob = 1.47312 (* 1 = 1.47312 loss)
8377 I0526 18:18:51.415205 1208 sgd_solver.cpp:105] Iteration 40, lr = 0.001
8378 I0526 18:19:04.153041 1221 data_layer.cpp:73] Restarting data prefetching from start.
8379 I0526 18:19:05.165007 1208 solver.cpp:239] Iteration 80 (2.9091 iter/s, 13.75s/40 iters), loss = 1.50252
1208 solver.cpp:258] Train net output #0: prob = 1.56949 (* 1 = 1.56949 loss)
8380 I0526 18:19:05.165084 1208 solver.cpp:258] Train net output #0: prob = 1.56949 (* 1 = 1.56949 loss)
8381 I0526 18:19:05.165092 1208 sgd_solver.cpp:105] Iteration 80, lr = 0.001
8382 I0526 18:19:18.839560 1208 solver.cpp:239] Iteration 120 (2.92512 iter/s, 13.6747s/40 iters), loss = 1.42754
1208 solver.cpp:258] Train net output #0: prob = 1.28254 (* 1 = 1.28254 loss)
8383 I0526 18:19:18.839609 1208 solver.cpp:258] Train net output #0: prob = 1.28254 (* 1 = 1.28254 loss)
8384 I0526 18:19:18.839712 1208 sgd_solver.cpp:105] Iteration 120, lr = 0.001
8385 I0526 18:19:32.218500 1221 data_layer.cpp:73] Restarting data prefetching from start.
8386 I0526 18:19:32.537020 1208 solver.cpp:239] Iteration 160 (2.92024 iter/s, 13.6975s/40 iters), loss = 1.45151
1208 solver.cpp:258] Train net output #0: prob = 1.35636 (* 1 = 1.35636 loss)
8387 I0526 18:19:32.537081 1208 solver.cpp:258] Train net output #0: prob = 1.35636 (* 1 = 1.35636 loss)
8388 I0526 18:19:32.537102 1208 sgd_solver.cpp:105] Iteration 160, lr = 0.001
8389 I0526 18:19:46.206521 1208 solver.cpp:239] Iteration 200 (2.92621 iter/s, 13.6696s/40 iters), loss = 1.39892
1208 solver.cpp:258] Train net output #0: prob = 1.06721 (* 1 = 1.06721 loss)
8390 I0526 18:19:46.206576 1208 solver.cpp:258] Train net output #0: prob = 1.06721 (* 1 = 1.06721 loss)
8391 I0526 18:19:46.206585 1208 sgd_solver.cpp:105] Iteration 200, lr = 0.001
8392 I0526 18:19:59.875075 1208 solver.cpp:239] Iteration 240 (2.9264 iter/s, 13.6687s/40 iters), loss = 1.38339
1208 solver.cpp:258] Train net output #0: prob = 1.5443 (* 1 = 1.5443 loss)
8393 I0526 18:19:59.875246 1208 solver.cpp:258] Train net output #0: prob = 1.5443 (* 1 = 1.5443 loss)
8394 I0526 18:19:59.875269 1208 sgd_solver.cpp:105] Iteration 240, lr = 0.001
8395 I0526 18:20:00.247403 1221 data_layer.cpp:73] Restarting data prefetching from start.
8396 I0526 18:20:13.555521 1208 solver.cpp:239] Iteration 280 (2.92388 iter/s, 13.6804s/40 iters), loss = 1.32526
1208 solver.cpp:258] Train net output #0: prob = 1.21835 (* 1 = 1.21835 loss)
8397 I0526 18:20:13.555572 1208 solver.cpp:258] Train net output #0: prob = 1.21835 (* 1 = 1.21835 loss)
8398 I0526 18:20:13.555580 1208 sgd_solver.cpp:105] Iteration 280, lr = 0.001
8399 I0526 18:20:27.289429 1208 solver.cpp:239] Iteration 320 (2.91248 iter/s, 13.734s/40 iters), loss = 1.32797
1208 solver.cpp:258] Train net output #0: prob = 1.77142 (* 1 = 1.77142 loss)
8400 I0526 18:20:27.289485 1208 solver.cpp:258] Train net output #0: prob = 1.77142 (* 1 = 1.77142 loss)

```

Figure 16: The log file of the model with the balanced dataset diminishing improving loss on the first few iterations as indicated by the red arrows, Mammo

## 2. Quiz Results

After the user has answered the questions and reviewed the answers, he/she may submit the quiz for evaluation by the "Submit" button seen in figure 26. The user's score and the correct answer are seen in figure 27.

```

mammo_resnet50_balanced_model.log [read-only] (~/.Desktop/caffe/models/mammo_resnet50_balanced) - gedit
9403 10526 19:07:35.447895 1208 solver.cpp:418] Test net output #18: probt = 0.268616
9404 10526 19:07:35.447898 1208 solver.cpp:418] Test net output #19: probt = 0.325682
9405 10526 19:07:38.508833 1208 solver.cpp:239] Iteration 8600 (2.8756 iter/s, 13.9101s/40 iters), loss = 0.000146464
9406 10526 19:07:38.508888 1208 solver.cpp:258] Train net output #0: prob = 0.000186151 (* 1 = 0.000186151 loss)
9407 10526 19:07:38.508911 1208 sgd_solver.cpp:105] Iteration 8600, lr = 0.000692534
9408 10526 19:07:40.229568 1221 data_layer.cpp:73] Restarting data prefetching from start.
9409 10526 19:07:52.092200 1208 solver.cpp:239] Iteration 8640 (2.94479 iter/s, 13.5833s/40 iters), loss = 0.000158482
9410 10526 19:07:52.092255 1208 solver.cpp:258] Train net output #0: prob = 8.88527e-05 (* 1 = 8.88527e-05 loss)
9411 10526 19:07:52.092278 1208 sgd_solver.cpp:105] Iteration 8640, lr = 0.000692534
9412 10526 19:08:05.681102 1208 solver.cpp:239] Iteration 8680 (2.94359 iter/s, 13.5888s/40 iters), loss = 0.00013274
9413 10526 19:08:05.681319 1208 solver.cpp:258] Train net output #0: prob = 0.00016929 (* 1 = 0.00016929 loss)
9414 10526 19:08:05.681344 1208 sgd_solver.cpp:105] Iteration 8680, lr = 0.000692534
9415 10526 19:08:08.078130 1221 data_layer.cpp:73] Restarting data prefetching from start.
9416 10526 19:08:19.265885 1208 solver.cpp:239] Iteration 8720 (2.94452 iter/s, 13.5846s/40 iters), loss = 0.000132953
9417 10526 19:08:19.265928 1208 solver.cpp:258] Train net output #0: prob = 0.000104163 (* 1 = 0.000104163 loss)
9418 10526 19:08:19.265935 1208 sgd_solver.cpp:105] Iteration 8720, lr = 0.000692534
9419 10526 19:08:32.849097 1208 solver.cpp:239] Iteration 8760 (2.94482 iter/s, 13.5832s/40 iters), loss = 0.000138264
9420 10526 19:08:32.849151 1208 solver.cpp:258] Train net output #0: prob = 0.000132472 (* 1 = 0.000132472 loss)
9421 10526 19:08:32.849159 1208 sgd_solver.cpp:105] Iteration 8760, lr = 0.000692534
9422 10526 19:08:35.930374 1221 data_layer.cpp:73] Restarting data prefetching from start.
9423 10526 19:08:46.434408 1208 solver.cpp:239] Iteration 8800 (2.94437 iter/s, 13.5852s/40 iters), loss = 0.000153959
9424 10526 19:08:46.434468 1208 solver.cpp:258] Train net output #0: prob = 0.000170378 (* 1 = 0.000170378 loss)
9425 10526 19:08:46.434492 1208 sgd_solver.cpp:105] Iteration 8800, lr = 0.000692534
9426 10526 19:09:00.016700 1208 solver.cpp:239] Iteration 8840 (2.94503 iter/s, 13.5822s/40 iters), loss = 0.000145803
9427 10526 19:09:00.016752 1208 solver.cpp:258] Train net output #0: prob = 6.70582e-05 (* 1 = 6.70582e-05 loss)
9428 10526 19:09:00.016760 1208 sgd_solver.cpp:105] Iteration 8840, lr = 0.000692534
9429 10526 19:09:03.784865 1221 data_layer.cpp:73] Restarting data prefetching from start.
9430 10526 19:09:13.600215 1208 solver.cpp:239] Iteration 8880 (2.94476 iter/s, 13.5835s/40 iters), loss = 0.000149988
9431 10526 19:09:13.600435 1208 solver.cpp:258] Train net output #0: prob = 0.000108068 (* 1 = 0.000108068 loss)
9432 10526 19:09:13.600457 1208 sgd_solver.cpp:105] Iteration 8880, lr = 0.000692534
9433 10526 19:09:27.300990 1208 solver.cpp:239] Iteration 8920 (2.91959 iter/s, 13.7005s/40 iters), loss = 0.000132032
9434 10526 19:09:27.301054 1208 solver.cpp:258] Train net output #0: prob = 0.000320126 (* 1 = 0.000320126 loss)
9435 10526 19:09:27.301076 1208 sgd_solver.cpp:105] Iteration 8920, lr = 0.000692534
9436 10526 19:09:31.847471 1221 data_layer.cpp:73] Restarting data prefetching from start.
9437 10526 19:09:37.391297 1208 solver.cpp:468] Snapshotting to binary proto file /home/kapitan/Desktop/caffe/models/mam
resnet_50_solver_iter_8950.caffemodel
9438 10526 19:09:37.612540 1208 sgd_solver.cpp:273] Snapshotting solver state to binary proto file /home/kapitan/Desktop/

```

Figure 17: The log file of the model with the balanced dataset diminishing improving loss on the last remaining iterations as indicated by the red arrows, Mammo

```

Predicting image: test/703.png | actual: 3 | prediction: 3 | correct | correct so far: 304
Num of correct: 304 / 703 | num of inccorect: 400

```

Figure 18: The accuracy of the model after balancing the dataset, Mammo

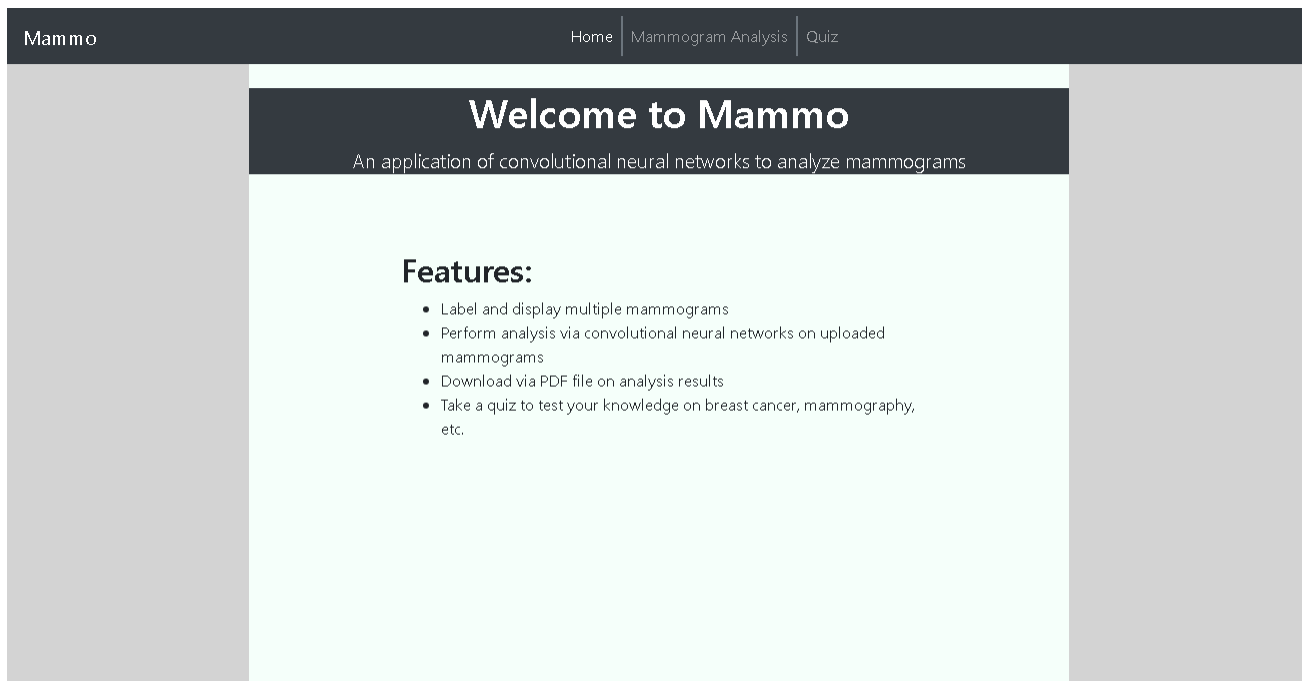


Figure 19: Mammo home page, Mammo

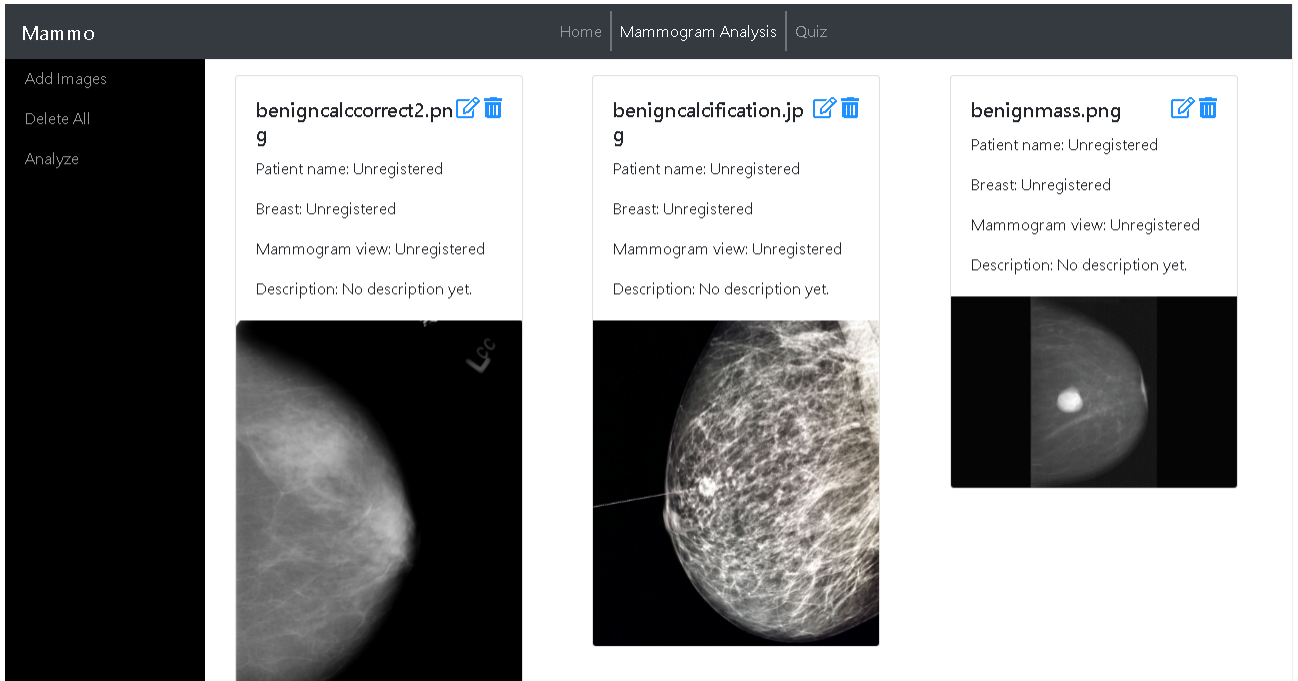


Figure 20: Three uploaded mammograms, Mammo

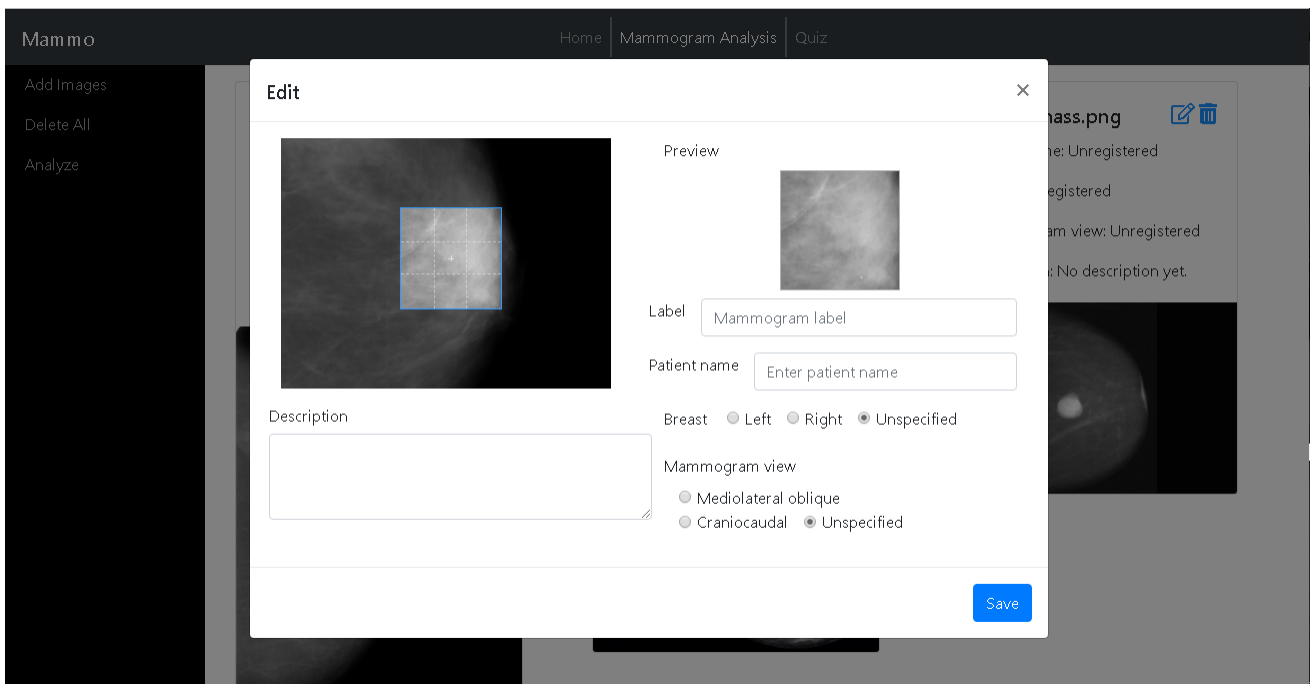


Figure 21: Editing metadata of the mammogram, Mammo



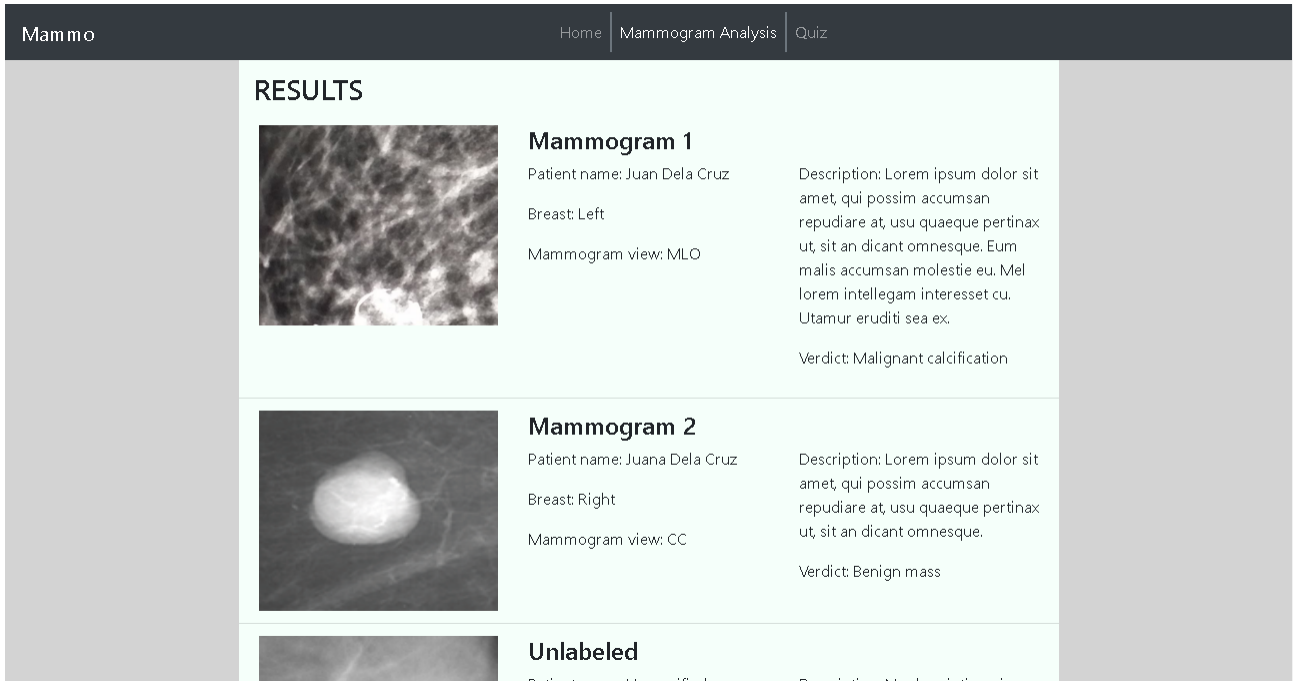


Figure 22: Results of the analysis made by the CNN module part 1, Mammo

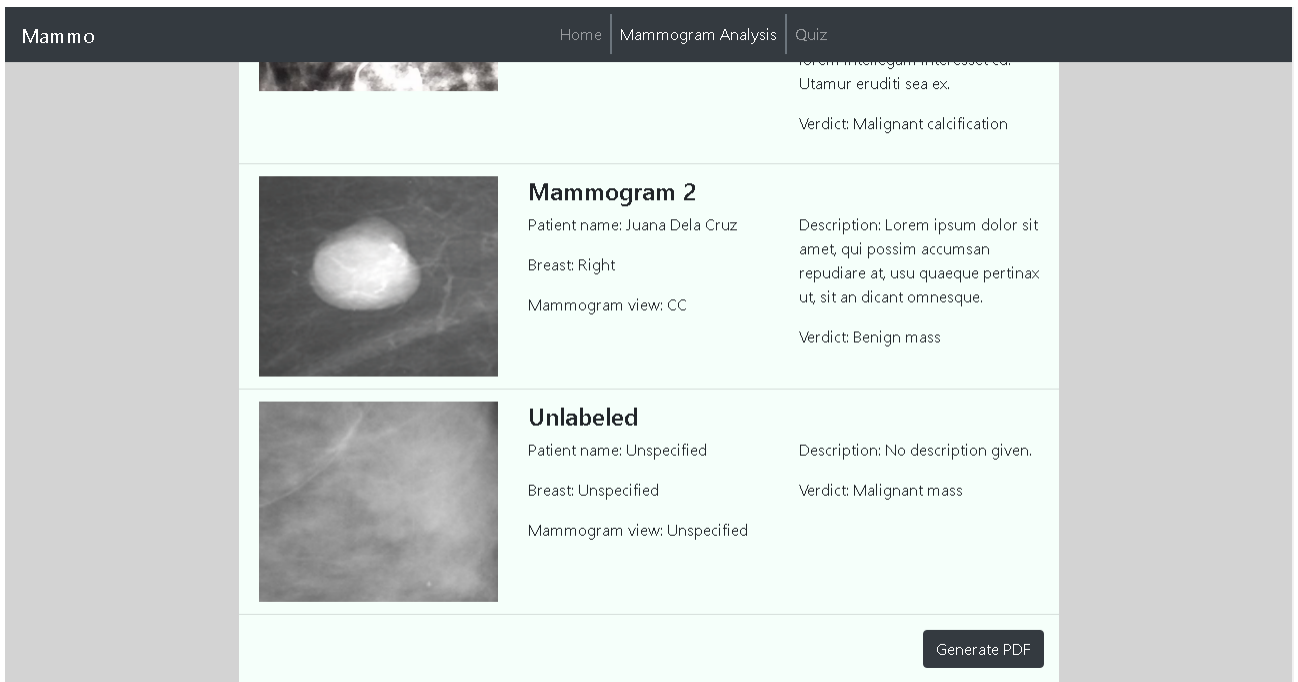


Figure 23: Results of the analysis made by the CNN module part 2, Mammo

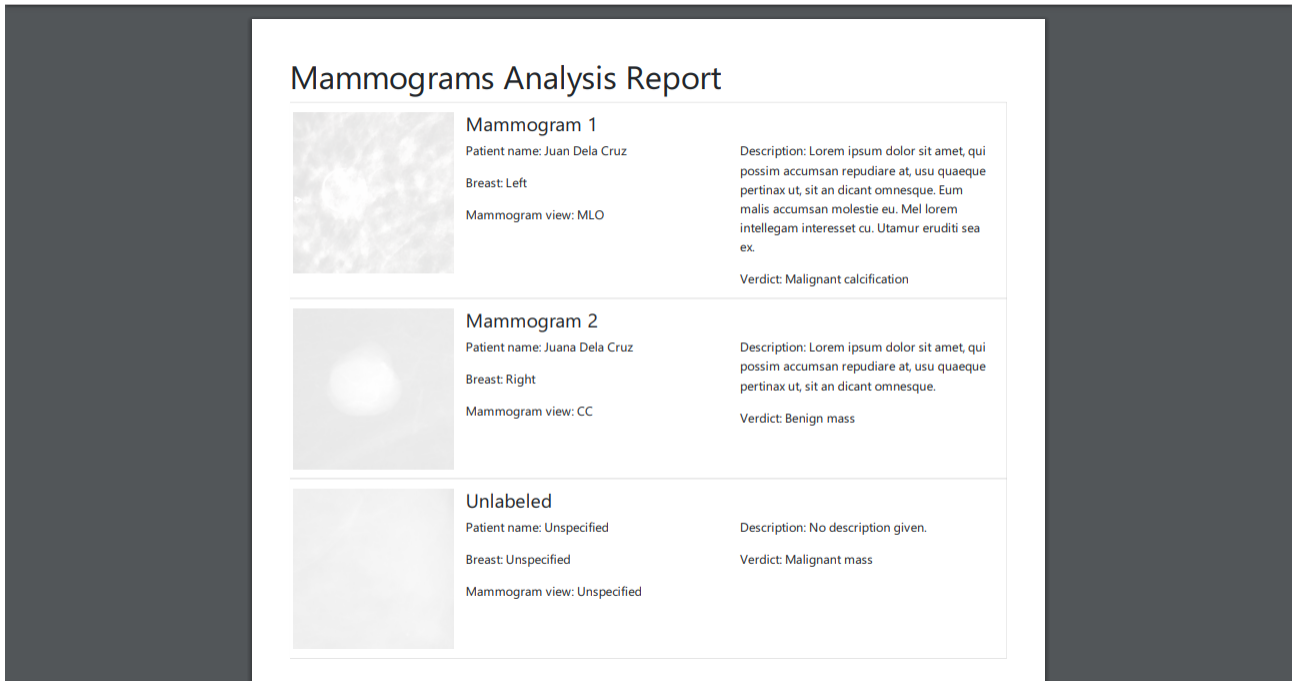


Figure 24: A generated PDF file of Mammogram Analysis Report, Mammo

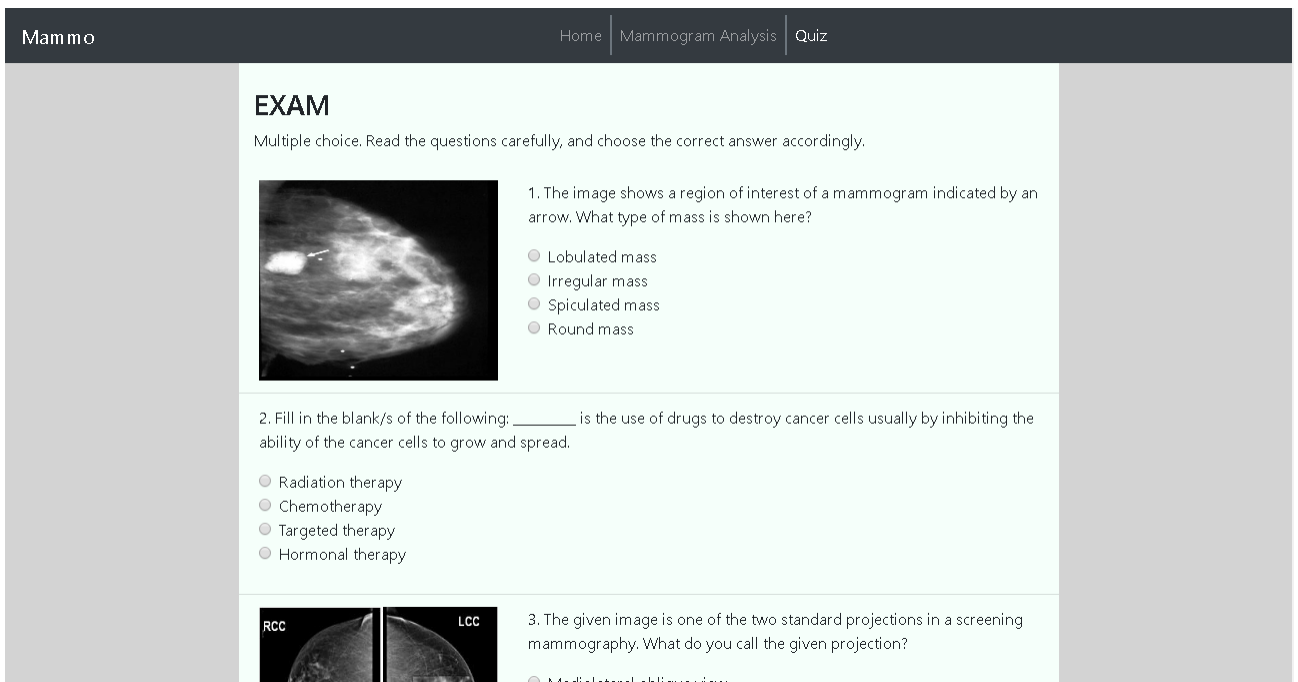
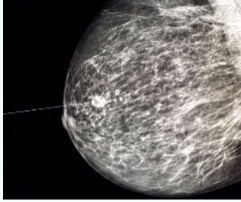


Figure 25: A sample of the quiz, Mammo

Mammo Home | Mammogram Analysis | Quiz

- Lobular carcinoma in situ
- Invasive ductal carcinoma
- Ductal carcinoma in situ

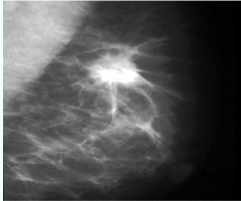
---



9. The image shows a mammogram with a large round region of interest indicated by the line. Is this a benign or malignant calcification?

- Benign calcification
- Malignant calcification

---



10. The given image is a type of mass that indicates malignancy. Which of the following types of mass is indicated here?

- Irregular masses
- Obscured masses
- Lobulated masses
- Spiculated masses

Figure 26: Submit the quiz, Mammo

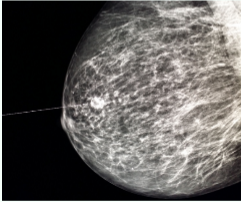
Mammo Home | Mammogram Analysis | Quiz

**TOTAL SCORE: 3**

1. Fill in the blank/s of the following: \_\_\_\_\_ targets the resources (proteins, specific genes, tissue environment) that fuel the survival and growth of cancer.

- Targeted therapy
- Chemotherapy
- Radiation therapy
- Hormonal therapy

---

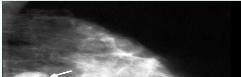


2. The image shows a mammogram with a large round region of interest indicated by the line. Is this a benign or malignant calcification?

- Benign calcification
- Malignant calcification
- 
- 

The correct answer is Benign calcification.

---



3. The image shows a region of interest of a mammogram indicated by an arrow. What type of mass is shown here?

Figure 27: Results of the quiz, Mammo

## VI. Discussion

Mammo is developed using a Convolutional Neural Network (CNN) specifically a Residual Neural Network with 50 layers which aims to assist radiologists, medical students and trainees to classify four breast lesions seen in mammography. It assists in classifying benign mass, malignant mass, benign calcification, and malignant calcification. The users have the functionality of uploading a mammogram, then choosing a region of interest for cropping and analysis by the model. The users also have the functionality of taking an exam in breast cancer, mammography, and other related topics.

The rationale behind having the user choose and crop a ROI from a mammogram is because the CNN model is purely trained on cropped ROIs of mammograms. Moreover, the model was trained in this manner because the ROIs have more distinct features for the model to work with when compared to their whole image counterparts [18].

The model was trained on CBIS-DDSM (Curated Breast Imaging Subset of the Digital Database for Screening Mammography). But, as stated before, the dataset was erroneous. There are several occurrences in the dataset's metadata where file locations are erroneous, so there is no way of confirming if the other information are erroneous as well. The proper measure was to clean the dataset by keeping track of the erroneous instances, and removing them from the dataset. This is because having a model trained under misclassified data would be problematic to the model's generalization. After the dataset has been cleaned, the images were converted from DICOM to PNG.

The hyperparameters of the model was set in the following manner: the number of epochs set to 40, the base learning rate set to 0.001, the batch size set to 179, and the optimizer set to Nesterov Gradient Descent. Initially, the optimizer was set to Stochastic Gradient Descent, but the model did not converge whereas when the optimizer was set to Nesterov Gradient Descent, the model slowly converged as indicated in the model's log file. However, after having the model tested, it only attained an accuracy of 33% as it correctly

labeled 243 images from a test set of 704. A possible cause for such a low accuracy is the lack of data augmentation techniques applied on the dataset used, for these are known to improve accuracy [24]. Another cause is the imbalanced number of instances per category specifically there are 681 ROI images for benign mass, 637 ROI images for malignant mass, 1,002 ROI images for benign calcification, and 544 ROI images for malignant calcification.

To address the low accuracy, the dataset was balanced having the number of images under each category set at a maximum of 544 ROI images. This was done by randomly removing extra images from each category until it reached 544 images except images under malignant calcification. Then the model was trained again with the same specifications, and the use of the balanced dataset. The accuracy improved up to 43% with the model correctly classifying 305 images out of 704.

In comparison with Shen's ResNet-50 model [18], the two networks are similar in terms of the architecture, and the dataset used. However, Shen utilized the whole-image version of the dataset while this study utilized the cropped region of interest (ROI) version of the dataset. Another notable difference was that Shen extracted the ROI from each of the whole images in the dataset and made a series of overlapping patches around the ROI in order to augment the dataset with the accuracy of the model reaching upwards of 80%. The model in this study only balanced the dataset in order to improve the accuracy.

The main advantage of using ResNet-50 over other CNN architectures such as AlexNet, GoogleNet, VGGNet, and other well-known models is that it outperforms them in terms of accuracy [18]. This is due to the number of layers that ResNet-50 has which is 50 whereas say AlexNet only has 8, for residual networks have the property of increasing their layers which would entail better model generalization without the vanishing or exploding gradients problem. Integrating this model into Mammo for mammogram classification makes Mammo useful as a means of providing a second opinion to radiologists when reading their mammograms, and as a means of training for medical trainees/students in interpreting mammograms.

## VII. Conclusions

Mammo is designed to provide radiologists and medical field students/trainees with a way to have mammograms analyzed and categorized under four breast lesion categories namely benign mass, malignant mass, benign calcification, and malignant calcification. Moreover, the software provides the medical students/trainees an exam to evaluate their knowledge regarding breast cancer, mammograms, etc.

The software utilized a Convolutional Neural Network, specifically ResNet-50, to analyze the mammograms' region of interest and to classify. Although initially the accuracy was poor, only achieving 33% with the test set, the accuracy was improved significantly up to 43% by balancing the dataset used in the development of the network. The data set used was from CBIS-DDSM (Curated Breast Imaging Subset of the Digital Database for Screening Mammography).

## VIII. Recommendations

Reflecting upon the training process, the model could be improved by applying data augmentation techniques such as applying a series of scaling, translations, rotations, and flipping on each image of the dataset consequently increasing the instances per category by a significant margin. It is also noted that the dataset should be made balanced for even higher accuracy.

Moreover, there should be an option for the users to retrain the model as they upload mammogram/s or to correct the model if it were to give out an erroneous reading as this would improve model accuracy and generalization well beyond initial training.

As for the design aspect of the model and application, having the model be trained on whole image mammograms would eliminate the need for the user to set a cropped ROI which is a tedious task especially when the user has a batch of mammograms to be analyzed.

## IX. Bibliography

- [1] Breastcancer.org, “What is breast cancer?” 2016. Available at [http://www.breastcancer.org/symptoms/understand\\_bc/what\\_is\\_bc](http://www.breastcancer.org/symptoms/understand_bc/what_is_bc).
- [2] WebMD, “What causes breast cancer?” 2017. Available at <https://www.webmd.com/breast-cancer/guide/what-causes-breast-cancer>.
- [3] Mayoclinic.org, “Breast cancer,” 2018. Available at <https://www.mayoclinic.org/diseases-conditions/breast-cancer/symptoms-causes/syc-20352470>.
- [4] Radiologyinfo.org, “Mammography,” 2017. Available at <https://www.radiologyinfo.org/en/info.cfm?pg=mammo>.
- [5] Breast cancer care, “Breast calcifications,” 2016. Available at <https://www.breastcancercare.org.uk/information-support/have-i-got-breast-cancer/benign-breast-conditions/breast-calcifications>.
- [6] Breastcancer.org, “Treatment & side effects,” 2017. Available at [http://www.breastcancer.org/treatment?gclid=EAIaIQobChMI69zvjp-12QIVyFe9Ch31tgpcEAAYAiAAEgKKuPD\\_BwE](http://www.breastcancer.org/treatment?gclid=EAIaIQobChMI69zvjp-12QIVyFe9Ch31tgpcEAAYAiAAEgKKuPD_BwE).
- [7] Breastcancer.org, “Breast cancer treatment,” 2017. Available at <https://www.webmd.com/breast-cancer/guide/breast-cancer-treatment#1>.
- [8] R. L. Siegel, K. D. Miller, and A. Jemal, “Cancer statistics, 2018,” *CA: A Cancer Journal for Clinicians*, vol. 68, no. 1, pp. 7–30, 2018. Available at <http://dx.doi.org/10.3322/caac.21442>.
- [9] M. Vardeleon, “Philippines has highest prevalence of breast cancer among 197 countries,” 2017. Available at <http://www.manilatimes.net/philippines-highest-prevalence-breast-cancer-among-197-countries/311287/>.



- [10] Philippine Daily Inquirer, “Incidence of breast cancer rising in ph, say experts,” 2017. Available at <http://newsinfo.inquirer.net/942804/philippine-news-updates-breast-cancer-dr-christina-galvez-philippine-breast-cancer->
- [11] C. D. Mangwang, “Cancer costs southeast asia socially and economically,” 2015. Available at <http://business.inquirer.net/198982/cancer-costs-southeast-asia-socially-and-economically>.
- [12] J. Mahanta, “Introduction to neural networks, advantages and applications,” 2017. Available at <https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207>.
- [13] S. Hijazi, R. Kumar, and C. Rowen, “Using convolutonal neural networks for image recognition,” 2015. Available at [https://ip.cadence.com/uploads/901/cnn\\_wp-pdf](https://ip.cadence.com/uploads/901/cnn_wp-pdf).
- [14] K. E. delas Penas, P. T. Rivera, and P. C. Naval, Jr., “Malaria parasite detection and species identification on thin blood smears using a convolutional neural network,” *IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, 2017. Available at <https://doi.org/10.1109/CHASE.2017.36>.
- [15] J. Chen, X. Qi, O. Tervonen, O. Silven, G. Zhao, and M. Pietikainen, “Thorax disease diagnosis using deep convolutional neural network,” *IEE*, 2017. Available at <https://doi.org/10.1109/EMBC.2016.7591186>.
- [16] S. Shichijo, S. Nomura, K. Aoyama, Y. Nishikawa, M. Miura, T. Shinagawa, H. Takiyama, T. Tanimoto, S. Ishihara, K. Matsuo, and T. Tada, “Application of convolutional neural networks in the diagnosis of helicobacter pylori infection based on endoscopic images,” *EBioMedicine*, vol. 25, pp. 106–111, 2017. Available at <https://doi.org/10.1016/j.ebiom.2017.10.014>.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *ARXIV*, 2015. Available at <https://arxiv.org/pdf/1512.03385.pdf>.

- [18] L. Shen, “End-to-end training for whole image breast cancer diagnosis using an all convolutional design,” *ARXIV*, 2017. Available at <https://arxiv.org/ftp/arxiv/papers/1708/1708.09427.pdf>.
- [19] J. A. Quinn, R. Nakasi, P. K. B. Mugagga, P. Byanyima, W. Lubega, and A. Andama, “Deep convolutional neural networks for microscopy-based point of care diagnostics,” *ARXIV*, 2016. Available at <https://arxiv.org/abs/1608.02989>.
- [20] C. Pearce, “Convolutional neural networks and the analysis of cancer imagery,” 2017. Available at <http://cs231n.stanford.edu/reports/2017/pdfs/25.pdf>.
- [21] K. J. Geras, S. Wolfson, Y. Shen, G. Kim, L. Moy, and K. Cho, “High-resolution breast cancer screening with multi-view deep convolutional neural networks,” *ARXIV*, 2017. Available at <https://arxiv.org/pdf/1703.07047.pdf>.
- [22] W. Sun, T.-L. Tseng, J. Zhang, and W. Qian, “Enhancing deep convolutional neural network scheme for breast cancer diagnosis with unlabeled data,” *Computerized Medical Imaging and Graphics*, vol. 57, pp. 4–9, 2017. Available at <https://doi.org/10.1016/j.compmedimag.2016.07.004>.
- [23] H. Chougrad, H. Zouaki, and O. Alheyane, “Deep convolutional neural networks for breast cancer screening,” *Computer Methods and Programs in Biomedicine*, vol. 157, pp. 19–30, 2018. Available at <https://doi.org/10.1016/j.cmpb.2018.01.011>.
- [24] T. Arajo, G. Aresta, E. Castro, J. Rouco, P. Aguiar, C. Eloy, A. Polnia, and A. Campilho, “Classification of breast cancer histology images using convolutional neural networks,” *PLOS*, 2017. Available at <https://doi.org/10.1371/journal.pone.0177544>.
- [25] A. Rakhlin, A. Shvets, V. Iglovikov, and A. A. Kalinin, “Deep convolutional neural networks for breast cancer histology image analysis,” *ARXIV*, 2018. Available at <https://arxiv.org/abs/1802.00752>.

- [26] S. Weng, "Automating breast cancer detection with deep learning," *INSIGHT*, 2017. Available at <https://blog.insightdatascience.com/automating-breast-cancer-detection-with-deep-learning-d8b49da17950>.
- [27] The American Cancer Society, "Types of breast cancer," 2017. Available at <https://www.cancer.org/cancer/breast-cancer/understanding-a-breast-cancer-diagnosis/types-of-breast-cancer.html>.
- [28] Breastcancer.org, "Dcis ductal carcinoma in situ," 2017. Available at <http://www.breastcancer.org/symptoms/types/dcis>.
- [29] The American Cancer Society, "Treatment of ductal carcinoma in situ (dcis)," 2016. <https://www.cancer.org/cancer/breast-cancer/treatment/treatment-of-breast-cancer-by-stage/treatment-of-ductal-carcinoma-in-situ-dcis.html>.
- [30] Breastcancer.org, "Idc invasive ductal carcinoma," 2017. Available at <http://www.breastcancer.org/symptoms/types/idc>.
- [31] Breastcancer.org, "Ilc invasive lobular carcinoma," 2017. Available at <http://www.breastcancer.org/symptoms/types/ilc>.
- [32] The American Cancer Society, "Breast cancer signs and symptoms," 2017. Available at <https://www.cancer.org/cancer/breast-cancer/about/breast-cancer-signs-and-symptoms.html>.
- [33] The American Cancer Society, "Breast cancer survival rates," 2017. Available at <https://www.cancer.org/cancer/breast-cancer/understanding-a-breast-cancer-diagnosis/breast-cancer-survival-rates.html>.
- [34] ASCO Cancer.net, "Breast cancer: Treatment options," 2017. Available at <https://www.cancer.net/cancer-types/breast-cancer/treatment-options>.
- [35] RadiologyInfo.org, "Breast cancer screening," 2016. Available at <https://www.radiologyinfo.org/en/info.cfm?pg=screening-breast>.

- [36] The American Cancer Society, “Breast biopsy,” 2017. Available at <https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/breast-biopsy.html>.
- [37] T. B. Bevers, B. O. Anderson, E. Bonaccio, S. Buys, M. B. Daly, P. J. Dempsey, W. B. Farrar, I. Fleming, J. E. Garber, R. E. Harris, A. S. Heerdt, M. Helvie, J. G. Huff, N. Khakpour, S. A. Khan, H. Krontiras, G. Lyman, E. Rafferty, S. Shaw, M. L. Smith, T. N. Tsangaris, C. Williams, and T. Yankeelov, “Breast cancer screening and diagnosis,” *Journal of the National Comprehensive Cancer Network*, vol. 7, no. 10, pp. 1083–1084, 2009. Available at <https://doi.org/10.1016/j.ebiom.2017.10.014>.
- [38] J. Heron, “Interpreting mammograms,” 2009. Available at [https://oncoprof.net/Generale2000/g04\\_Diagnostic/Mammographie/gb04\\_mm04.html](https://oncoprof.net/Generale2000/g04_Diagnostic/Mammographie/gb04_mm04.html).
- [39] J. Heron, “Breast masses,” 2009. Available at [https://oncoprof.net/Generale2000/g04\\_Diagnostic/Mammographie/gb04\\_mm06.html](https://oncoprof.net/Generale2000/g04_Diagnostic/Mammographie/gb04_mm06.html).
- [40] J. Heron, “Regular breast masses,” 2009. Available at [https://oncoprof.net/Generale2000/g04\\_Diagnostic/Mammographie/gb04\\_mm07.html](https://oncoprof.net/Generale2000/g04_Diagnostic/Mammographie/gb04_mm07.html).
- [41] J. Heron, “Masses with irregular limits,” 2009. Available at [https://oncoprof.net/Generale2000/g04\\_Diagnostic/Mammographie/gb04\\_mm08.html](https://oncoprof.net/Generale2000/g04_Diagnostic/Mammographie/gb04_mm08.html).
- [42] J. Heron, “Irregular masses,” 2009. Available at [https://oncoprof.net/Generale2000/g04\\_Diagnostic/Mammographie/gb04\\_mm09.html](https://oncoprof.net/Generale2000/g04_Diagnostic/Mammographie/gb04_mm09.html).
- [43] WebMD, “Breast calcifications,” 2017. Available at <https://www.webmd.com/women/guide/breast-calcification-symptoms-causes-treatments#1>.
- [44] J. Heron, “Mammary calcifications,” 2009. Available at [https://oncoprof.net/Generale2000/g04\\_Diagnostic/Mammographie/gb04\\_mm10.html](https://oncoprof.net/Generale2000/g04_Diagnostic/Mammographie/gb04_mm10.html).
- [45] D. Faggella, “What is machine learning,” 2017. Available at <https://www.techemergence.com/what-is-machine-learning/>.

- [46] D. Fumo, “Types of machine learning algorithms you should know,” 2017. Available at <https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861>.
- [47] V. Maini, “Machine learning for humans, part 2.1: Supervised learning,” 2017. Available at <https://medium.com/machine-learning-for-humans/supervised-learning-740383a2feab>.
- [48] D. Shiffman, “The nature of code: Chapter 10. neural networks,” 2012. Available at <http://natureofcode.com/book/chapter-10-neural-networks/>.
- [49] O. Maslovska, “Deep learning: Definition, benefits, and challenges,” 2017. Available at <https://stfalcon.com/en/blog/post/deep-learning-benefits-and-challenges>.
- [50] S. Agatonovic-Kustrin and R. Beresford, “Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research,” *Journal of Pharmaceutical and Biomedical Analysis*, vol. 22, pp. 717–727, 2000. Available at [https://doi.org/10.1016/S0731-7085\(99\)00272-1](https://doi.org/10.1016/S0731-7085(99)00272-1).
- [51] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” 2017. Available at [https://www.mitpressjournals.org/doi/pdf/10.1162/neco\\_a\\_00990](https://www.mitpressjournals.org/doi/pdf/10.1162/neco_a_00990).
- [52] “Cs231n convolutional neural networks for visual recognition.” Available at <http://cs231n.github.io/convolutional-networks/#conv>.
- [53] R. S. Lee, F. Gimenez, A. Hoogi, and D. Rubin, “Curated breast imaging subset of ddsn,” 2016. The Cancer Imaging Archive. Available at <http://dx.doi.org/10.7937/K9/TCIA.2016.7002S9CY>.
- [54] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, “The cancer imaging archive (tcia): Maintaining and operating a public information repository,” *Journal of Digital Imag-*

*ing*, vol. 26, no. 6, pp. 1045–1057, 2013. Available at <https://link.springer.com/article/10.1007%2Fs10278-013-9622-7>.

## X. Appendix

### *app.py*

```
import os, json
import numpy as np
import jinja2
import random
import pdfkit

from subprocess import call
from flask import Flask, render_template, request,
    send_from_directory
from flask import Response, make_response
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy.sql.expression import func
from werkzeug import secure_filename
from datetime import datetime

APP_ROOT = os.path.dirname(os.path.abspath(__file__))
UPLOAD_FOLDER = os.path.join(APP_ROOT, 'uploads/')
CROPPED_IMAGE_FOLDER = os.path.join(APP_ROOT, '
    croppedImages/')
ALLOWED_EXTENSIONS = set(['png', 'jpg', 'jpeg'])

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///
    mammograms.sqlite3'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
app.config['CROPPED_IMAGE_FOLDER'] =
    CROPPED_IMAGE_FOLDER
app.config['SESSION_ID'] = str(datetime.now())
app.jinja_env.filters['zip'] = zip

meanFile = './caffe-windows/python/mammogram_mean.npy'
deploy = './caffe-windows/python/resnet-deploy.prototxt'
caffeModel = './caffe-windows/python/
    resnet_50_solver_iter_7160.caffemodel'
classify = './caffe-windows/python/classify.py'
cropped_images = app.config['CROPPED_IMAGE_FOLDER']
pred_result = './result.npy'

db = SQLAlchemy(app)

class mammograms(db.Model):
    id = db.Column('file.id', db.Integer, primary_key = True)
    sessionId = db.Column(db.String(50))
    fileName = db.Column(db.String(50))
    fileLocation = db.Column(db.String(100))
    title = db.Column(db.String(50))
    patientName = db.Column(db.String(50))
    breast = db.Column(db.String(10))
    breastView = db.Column(db.String(10))
    notes = db.Column(db.String(100))
    croppedFile = db.Column(db.String(100))

def __init__(self, sessionId, fileName, fileLocation, title,
    patientName, breast, breastView, notes, croppedFile):
    self.sessionId = sessionId
    self.fileName = fileName
    self.fileLocation = fileLocation
    self.title = title
    self.patientName = patientName
    self.breast = breast
    self.breastView = breastView
    self.notes = notes
    self.croppedFile = croppedFile

class quizItems(db.Model):
    id = db.Column('question.id', db.Integer, primary_key = True)
    question = db.Column(db.String(100))
    questionImage = db.Column(db.String(50))
    answer = db.Column(db.String(30))
    otherChoice1 = db.Column(db.String(30))
    otherChoice2 = db.Column(db.String(30))
    otherChoice3 = db.Column(db.String(30))

def __init__(self, question, questionImage, answer,
    otherChoice1,
    otherChoice2, otherChoice3):
    self.question = question
    self.questionImage = questionImage
    self.answer = answer
    self.otherChoice1 = otherChoice1
    self.otherChoice2 = otherChoice2
    self.otherChoice3 = otherChoice3

@app.route('/')
def home():
    navbarActive = 'home'
    return render_template('home.html', navbar=navbarActive)

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    listMammograms = mammograms.query.all()
    for mammogram in listMammograms:
        imageToDelete = mammogram.fileName
        os.remove(os.path.join(app.config['UPLOAD_FOLDER'],
            imageToDelete))
        croppedImageToDelete = mammogram.croppedFile
        if croppedImageToDelete != '':
            os.remove(os.path.join(app.config['
                CROPPED_IMAGE_FOLDER'],
                croppedImageToDelete))
        db.session.delete(mammogram)
        db.session.commit()

    if request.method == 'POST':
        errorMessage = ''
```

```

if 'file' not in request.files:
    errorMessage = 'No file uploaded.'
    return Response(json.dumps(errorMessage), status=500,
                    mimetype='application/json')

imageGalleryRender = ''

for file in request.files.getlist("file"):
    if file.filename == '':
        errorMessage = 'No filename uploaded.'
        return Response(json.dumps(errorMessage), status=500,
                        mimetype='application/json')

    if file and isFileAllowed(file.filename):
        sessionId = app.config['SESSION_ID']
        filename = secure_filename(file.filename)
        fileLocation = 'predict/' + filename
        title = filename
        patientName = 'Unregistered'
        breast = 'Unregistered'
        breastView = 'Unregistered'
        notes = 'No description yet.'
        croppedFile = ''
        file.save("".join([app.config['UPLOAD_FOLDER'],
                           filename]))
        mammogram = mammograms(sessionId, filename,
                                fileLocation,
                                title, patientName, breast, breastView, notes, croppedFile)
        db.session.add(mammogram)
        db.session.commit()

    id = mammograms.query.filter_by(fileName=filename).first().id

    imageGalleryRender +=
'<div style="width: 200px;" class="col-lg-4 col-md-5 col
  -xs-6 mx-lg-0 ' \
'mx-md-5 mx-sm-6 mb-lg-4 mb-md-3 mb-sm-2
  container thumbnail-container" ' \
'thumbnail-id="' + str(id) + '" thumbnail-src="predict/'
  + filename + '"> \n' \
' <div class="card" style="width: 18rem;"> \n' \
' <div class="card-body"> \n' \
' <span class="modal-icon delete-icon d-inline float-
  right" ' \
' data-toggle="tooltip" data-placement="top" title="
  Delete"> \n' \
' <i class="fas fa-trash-alt fa-lg"></i> \n' \
' </span> \n' \
' <span class="modal-icon mr-1 edit-icon d-inline
  float-right" ' \
' data-target="#modal" data-toggle="modal" data-
  toggle="tooltip" ' \
' data-placement="top" title="Edit"> \n' \
' <i class="far fa-edit fa-lg"></i> \n' \
' </span> \n' \
' <h5 id="card_' + str(id) + '_title" class="card-title
  ' > ' +
  filename + '</h5> \n' \
' <p id="card_' + str(id) + '_patientName" class="card-
  text">' \
' Patient name: ' + patientName + '</p> \n' \
' <p id="card_' + str(id) + '_breast" class="card-text
  ">' \
' Breast: ' + breast + '</p> \n' \
' <p id="card_' + str(id) + '_breastView" class="card-
  text">' \
' Mammogram view: ' + breastView + '</p> \n' \
' <p id="card_' + str(id) + '_notes" class="card-text">
  Description: ' + notes + '</p> \n' \
' </div> \n' \
' <a href="" class="d-block thumbnail" style="z-index:
  3;" data-target="#modal" ' \
' data-toggle="modal"> \n' \
' <div class="thumbnail-img-container" data-
  imageSource="predict/id=' + str(id) + '"> \n' \
' <img class="card-img-bottom img-responsive" src
  ="predict/'
  + filename + '" alt="Card image cap" /> \n' \
' <div class="overlay"> \n' \
'  \n' \
' </div> \n' \
' </a> \n' \
' </div> \n'
'</div> \n'
else:
    errorMessage = 'Only .png, .jpg, and .jpeg files are
    accepted.'
    return Response(json.dumps(errorMessage), status=500,
                    mimetype='application/json')

return Response(json.dumps(imageGalleryRender), mimetype
                ='application/json')

navbarActive = 'predict'
return render_template('predict.html', navbar=navbarActive)

@app.route('/predict/id=<id>')
def sendImageById(id):
    filename = mammograms.query.filter_by(id=id).first().fileName
    return sendImage(filename)

@app.route('/predict/<filename>')
def sendImage(filename):
    return send_from_directory('uploads', filename)

@app.route('/delete', methods=['POST'])
def deleteMammogram():
    idMammogram = request.form['id']
    mammogram = mammograms.query.filter_by(id=idMammogram
    ).first()
    imageToDelete = mammogram.fileName
    os.remove(os.path.join(app.config['UPLOAD_FOLDER'],

```



```

        imageToDelete)
croppedImageToDelete = mammogram.croppedFile
if croppedImageToDelete != '':
    os.remove(os.path.join(app.config['
        CROPPED_IMAGE_FOLDER'], croppedImageToDelete))
db.session.delete(mammogram)
db.session.commit()
return Response(json.dumps('success'), mimetype='application
    /json')

@app.route('/deleteAll', methods=['POST'])
def deleteAll():
    listMammograms = mammograms.query.all()
    for mammogram in listMammograms:
        imageToDelete = mammogram.fileName
        os.remove(os.path.join(app.config['UPLOAD_FOLDER'],
            imageToDelete))
        croppedImageToDelete = mammogram.croppedFile
        if croppedImageToDelete != '':
            os.remove(os.path.join(app.config['
                CROPPED_IMAGE_FOLDER'], croppedImageToDelete)
            )
        db.session.delete(mammogram)
        db.session.commit()
    return Response(json.dumps('success'), mimetype='application
        /json')

@app.route('/saveMammogram', methods=['POST'])
def saveMammogram():
    idMammogram = request.form['id']
    mammogram = mammograms.query.filter_by(id=idMammogram
        ).first()
    mammogram.title = request.form['title']
    if mammogram.title == '':
        mammogram.title = 'Unlabeled'
    mammogram.patientName = request.form['patientName']
    if mammogram.patientName == '':
        mammogram.patientName = 'Unspecified'
    mammogram.breast = request.form['breast']
    mammogram.breastView = request.form['breastView']
    mammogram.notes = request.form['notes']
    db.session.commit()
    mammogram = mammograms.query.filter_by(id=idMammogram
        ).first()
    return Response(json.dumps('Updated successfully'), mimetype
        ='application/json')

@app.route('/saveCroppedFile', methods=['POST'])
def saveCroppedFile():
    file = request.files['croppedFile']
    idMammogram = request.form['id']
    mammogram = mammograms.query.filter_by(id=idMammogram
        ).first()
    fileName = mammogram.fileName.split('.')[0] + '_' + file.name
        + '_' + str(idMammogram) + '.png'
    file.save("".join([app.config['CROPPED_IMAGE_FOLDER'],
        fileName]))

mammogram = mammograms.query.filter_by(id=idMammogram
    ).first()
mammogram.croppedFile = fileName
db.session.commit()
return Response(json.dumps('Updated successfully'), mimetype
    ='application/json')

@app.route('/predictResults')
def predictResults():
    listMammograms = mammograms.query.all()
    navbarActive = 'predict'

    call(['python', classify, cropped_images, pred_result, '--
        model_def', deploy,
        '--pretrained_model', caffeModel, '--mean_file', meanFile,
        '--center_only'])

    res = []
    data = np.load('result.npy')
    for d in data:
        predMade = np.argmax(d)
        if predMade == 0:
            res.append('Benign mass')
        elif predMade == 1:
            res.append('Malignant mass')
        elif predMade == 2:
            res.append('Benign calcification')
        else:
            res.append('Malignant calcification')

    return render_template('predictResults.html', mammograms=
        listMammograms,
        navbar=navbarActive, result=res)

@app.route('/predictResults/<filename>')
def sendCroppedFile(filename):
    croppedImageToDelete = filename
    return send_from_directory('croppedImages', filename)

@app.route('/pdfPredictResults')
def pdfPredictResults():
    listMammograms = mammograms.query.all()
    res = []
    data = np.load('result.npy')
    for d in data:
        predMade = np.argmax(d)
        if predMade == 0:
            res.append('Benign mass')
        elif predMade == 1:
            res.append('Malignant mass')
        elif predMade == 2:
            res.append('Benign calcification')
        else:
            res.append('Malignant calcification')
    path_wkhtmltopdf = r'C:\Program Files\wkhtmltopdf\bin\
        wkhtmltopdf.exe'
    config = pdftk.configuration(wkhtmltopdf=path_wkhtmltopdf)

```

```

rendered = render_template('pdfPredictResults.html',
    mammograms=listMammograms, result=res,
    filepath=app.config['CROPPED_IMAGE_FOLDER'])
css = ['./static/css/bootstrap.min.css', './static/css/
    pdfResults.css']
pdf = pdfkit.from_string(rendered, False, configuration=config,
    css=css)
response = make_response(pdf)
response.headers['Content-Type'] = 'application/pdf'
response.headers['Content-Disposition'] = 'inline; filename=
    output.pdf'
return response

```

```

@app.route('/quiz', methods=['GET'])
def quiz():
    quiz = quizItems.query.order_by(func.random()).limit(10).all()
    quizData = []
    navbarActive = 'quiz'
    i = 0
    for quizItem in quiz:
        otherChoices = []
        quizId = quizItem.id
        otherChoices.append(quizItem.answer)
        otherChoices.append(quizItem.otherChoice1)
        otherChoices.append(quizItem.otherChoice2)
        otherChoices.append(quizItem.otherChoice3)
        if quizItem.otherChoice2 != '':
            random.shuffle(otherChoices)
        quizDataItem = {'id': quizId, 'question': quizItem.question,
            'questionImage': quizItem.questionImage, 'correctAnswer':
                quizItem.answer,
            'otherChoices': otherChoices}
        quizData.append(quizDataItem)
    return render_template('quiz.html', quizItems=quizData,
        navbar=navbarActive)

```

```

@app.route('/quizResults', methods=['POST'])
def quizResults():
    score = 0
    quizResultsDataForRender = []
    for i in range(0, 10):
        id = request.form['id' + str(i)]
        quizItem = quizItems.query.filter_by(id=id).first()
        question = quizItem.question
        questionImage = quizItem.questionImage
        correctAnswer = quizItem.answer
        otherChoices = []
        otherChoices.append(quizItem.answer)
        otherChoices.append(quizItem.otherChoice1)
        otherChoices.append(quizItem.otherChoice2)
        otherChoices.append(quizItem.otherChoice3)
        if quizItem.otherChoice2 != '':
            random.shuffle(otherChoices)
        answer = request.form['answer' + str(i)]
        if quizItem.answer == request.form['answer' + str(i)]:
            result = 'Correct'
            score += 1

```

```

else:
    result = 'Incorrect'
    i += 1
    quizResultItem = {'questionRender': question, 'questionImage
        ': questionImage,
        'correctAnswer': correctAnswer, 'otherChoices': otherChoices,
        'userAnswer': answer,
        'quizResult': result}
    quizResultsDataForRender.append(quizResultItem)
    navbarActive = 'quiz'
    return render_template('quizResults.html', navbar=
        navbarActive, quizResultsData=
            quizResultsDataForRender,
        finalScore=score)

```

```

def isFileAllowed(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower
        () in ALLOWED_EXTENSIONS

```

```

if __name__ == '__main__':
    db.create_all()
    app.run()

```

## utilities.py

```

import subprocess

def convertBinaryprotoToNpy():
    convert = './caffe-windows/python/convert.py'
    proto = './caffe/data/mammo_resnet_50/mammogram_mean.
        binaryproto'
    npy = './caffe/data/mammo_resnet_50/mammogram_mean.npy'
    call(['python', convert, proto, npy])

def trainModel(solver, log, parse):
    subprocess.Popen(['git-bash', 'train.sh', solver, log])

def predict(input, output, meanFile, deploy, model)
    classify = './caffe-windows/python/classify.py'
    call(['python', classify, input, output, '--model_def', deploy
        , '--pretrained_model', model, '--mean_file', meanFile,
        '--center_only'])

```

## templates/includes/navbar.html

```

<nav class="navbar navbar-expand-md navbar-dark bg-dark
    fixed-top">
    <a class="navbar-brand" href="{{ url_for('home') }}">
        Mammo</a>
    <button class="navbar-toggler" type="button" data-
        toggle="collapse" data-target="#
            navbarsExampleDefault" aria-controls="
                navbarsExampleDefault" aria-expanded="false"
                aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>

```

```

<div class="collapse navbar-collapse" id="
  navbarsExampleDefault">
  <ul class="navbar-nav mx-auto">
{% if navbar == 'home' %}
  <li class="border-right border-secondary nav-item active
    ">
    <a class="nav-link" href="{ url_for('home') }" >
      Home<span class="sr-only">(current)</
        span></a>
{% else %}
  <li class="border-right border-secondary nav-item">
    <a class="nav-link" href="{ url_for('home') }" >
      Home</a>
{% endif %}
  </li>
  {% if navbar == 'predict' %}
  <li class="border-right border-left border-secondary nav
    -item active">
    <a class="nav-link" href="{ url_for('predict')
      }" >Mammogram Analysis<span class="sr-
        only">(current)</span></a>
{% else %}
  <li class="border-right border-left border-secondary nav
    -item">
    <a class="nav-link" href="{ url_for('predict')
      }" >Mammogram Analysis</a>
{% endif %}
  </li>
  {% if navbar == 'quiz' %}
  <li class="border-left border-secondary nav-item active
    ">
    <a class="nav-link" href="{ url_for('quiz') }" >
      Quiz<span class="sr-only">(current)</span
        ></a>
{% else %}
  <li class="border-left border-secondary nav-item">
    <a class="nav-link" href="{ url_for('quiz') }" >
      Quiz</a>
{% endif %}
  </li>
</ul>
</div>
</nav>

```

### templates/includes/sidebar.html

```

<div id="sidebar-wrapper">
<ul class="sidebar-nav">
<li>
  <input style="display:none" name="uploadImagesForm"
    type="file" id="uploadForm" multiple />
  <a href="#" id="uploadImages">Add Images</a>
</li>
<!--
<li>
  <a href="#">Search</a>
</li>

```

```

<li>
  <a href="#">Delete Images</a>
</li>
--
<li>
  <a id="deleteAll" href="#">Delete All</a>
</li>
<li>
  <a href="{ url_for('predictResults') }" >Analyze</a>
</li>
</ul>
</div>

```

### templates/layout.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge
    ">
  <meta name="viewport" content="width=device-width,
    initial-scale=1, shrink-to-fit=no">
  <title>{% block title %}Mammo{% endblock %}</title>
  <link rel="stylesheet" href="/static/css/bootstrap.min.css">
  <link rel="stylesheet" href="/static/css/fontawesome-all.css
    ">
  <link rel="stylesheet" href="/static/css/sidebar.css">
</head>
{% block head %}{% endblock %}
</head>
<body>
  {% include 'includes/navbar.html' %}
  {% block body %}{% endblock %}
  <script src="/static/js/jquery.js"></script>
  <script src="/static/js/popper.js"></script>
  <script src="/static/js/bootstrap.min.js"></script>
  {% block script %}{% endblock %}
</body>
</html>

```

### templates/home.html

```

{% extends 'layout.html' %}
{% block head -%}
  <link rel="stylesheet" href="/static/css/predictResults.css">
{% - endblock %}
{% block body %}
  <div id="resultsContainer" class="container predictResults-
    container px-0">
  <header style="position:relative; top:80px;" class="bg-dark
    text-white">
    <div class="container text-center">
    <h1>Welcome to Mammo</h1>
    <p class="lead">An application of convolutional neural
      networks to analyze mammograms</p>

```

```

</div>
</header>

<section style="position: relative; top: 140px;" id="about">
  <div class="container">
    <div class="row">
      <div class="col-lg-8 mx-auto">
        <h2>Features:</h2>
        <ul>
          <li>Label and display multiple mammograms</li>
          <li>Perform analysis via convolutional neural networks on
            uploaded mammograms</li>
          <li>Download via PDF file on analysis results</li>
          <li>Take a quiz to test your knowledge on breast cancer,
            mammography, etc.</li>
        </ul>
      </div>
    </div>
  </div>
</div>
</section>
</div>

{% endblock %}

{% block script %}
{%- endblock %}

```

### *templates/pdfPredictResults.html*

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
      initial-scale=1, shrink-to-fit=no">
  </head>
  <body>
    <h1>Mammograms Analysis Report</h1>
    <ul id="PDFcontent" class="list-group list-group-flush">
      {% for i in range(mammograms|length) %}
        <li class="list-group-item container">
          <div class="row">
            
            <div class="col">
              <h4>{{ mammograms[i].title }}</h4>
              <div class="row">
                <div class="col">
                  <p>Patient name: {{ mammograms[i].patientName
                    }}</p>
                  <p>Breast: {{ mammograms[i].breast }}</p>
                  <p>Mammogram view: {{ mammograms[i].breastView
                    }}</p>
                </div>
              </div>
            </div>
          </div>
        </li>
      {% endfor %}
    </ul>
  </body>
</html>

```

```

<p>Description: {{ mammograms[i].notes }}</p>
<p>Verdict: {{ result[i] }}</p>
</div>
</div>
</div>
</li>
{% endfor %}
</ul>
</body>
</html>

```

### *templates/predict.html*

```

{% extends 'layout.html' %}

{% block head %}
  <link rel="stylesheet" href="/static/css/cropper.css">
  <link rel="stylesheet" href="/static/css/predict.css">
{%- endblock %}

{% block body %}
  {% include 'includes/sidebar.html' %}
  <div style="position: fixed; top: 0; left: 0; z-index:
    999999999; width: 100%; height: 100%; background-
    color: rgba(0,0,0,0.5); visibility: hidden; opacity: 0;" class
    ="dropzone">
  </div>
  <div class="container-fluid float-left" style="max-width:
    100%;">
    <div class="row">
      <div class="col-2"></div>
      <div id="imageGalleryContainer" style="margin-top: 80px;
        ;" class="col">
        <div id="imageGallery" class="row text-center text-lg-
          left"></div>
      </div>
    </div>
  </div>

  <div class="modal fade" id="modal" tabindex="-1" role="
    dialog" aria-labelledby="modalLabel" aria-hidden="
    true">
    <div class="modal-dialog modal-lg modal-dialog-centered
      " role="document">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title" id="modalLabel">Edit</h5>
          <button type="button" class="close" data-dismiss="modal"
            aria-label="Close">
            <span aria-hidden="true">&times;</span>
          </button>
        </div>
        <div class="modal-body">
          <div class="container">
            <div class="row">
              <div class="col">
                <div class="img-container" style="height: 250px; width:

```

```

    330px;">
<img id="modalImage" style="border: 1px; max-width
    : 100%; visibility: hidden;" src="" alt="Picture">
</div>
</div>
<div class="col">
<label for="divPreview">Preview</label>
<div id="divPreview" class="preview mb-2" style="
    width: 120px; height:120px; border: 1px solid rgb
    (178,178,178); overflow: hidden; margin: 0 auto; text
    -align: center; visibility: hidden;">
</div>
<form id="mammogramData">
<input type="hidden" id="idMammogram" name="id"
    value="">
<div class="form-group row">
<label for="title">Label</label>
<div class="col">
<input type="text" class="form-control" name="title"
    id="title" placeholder="Mammogram label">
</div>
</div>
<div class="form-group row">
<label for="patientName">Patient name</label>
<div class="col">
<input type="text" class="form-control" name="
    patientName" id="patientName" placeholder="
    Enter patient name">
</div>
</div>
<div class="form-group row">
<legend class="col-form-label col-sm-2 pt-0">Breast
</legend>
<div class="col-sm-10">
<div class="form-check form-check-inline">
<input class="form-check-input" type="radio"
    name="breast" id="breastLeft" value="Left">
<label class="form-check-label" for="breastLeft">
    Left</label>
</div>
<div class="form-check form-check-inline">
<input class="form-check-input" type="radio"
    name="breast" id="breastRight" value="Right
    ">
<label class="form-check-label" for="breastRight
    ">Right</label>
</div>
<div class="form-check form-check-inline">
<input class="form-check-input" type="radio"
    name="breast" id="breastUnspecified" value="
    Unspecified" checked>
<label class="form-check-label" for="
    breastUnspecified">Unspecified</label>
</div>
</div>
</div>
<div class="form-group">

```

```

<label for="breastView">Mammogram view</label>
<div class="col-sm-10" id="breastView">
<div class="form-check form-check-inline">
<input class="form-check-input" type="radio"
    name="breastView" id="breastViewMLO" value
    ="MLO">
<label class="form-check-label" for="
    breastViewMLO">Mediolateral oblique</label>
</div>
<div class="form-check form-check-inline">
<input class="form-check-input" type="radio"
    name="breastView" id="breastViewCC" value
    ="CC">
<label class="form-check-label" for="breastViewCC
    ">Craniocaudal</label>
</div>
<div class="form-check form-check-inline">
<input class="form-check-input" type="radio"
    name="breastView" id="breastViewUnspecified"
    value="Unspecified" checked>
<label class="form-check-label" for="
    breastViewUnspecified">Unspecified</label>
</div>
</div>
</div>
<div class="form-group">
<label style="position:absolute; left : -380px; top: 265
    px;" for="notes">Description</label>
<textarea style="position:absolute; left : -380px; top:
    295px;" class="form-control" id="notes" name="
    notes" rows="3"></textarea>
</div>
</form>
</div>
</div>
<div class="modal-footer">
<button id="saveMammogram" type="button" data-dismiss
    ="modal" class="btn btn-primary">Save</button>
</div>
</div>
</div>
{% endblock %}
{% block script -%}
<script type="text/javascript">
$(document).ready(function(){
    $('[data-toggle="tooltip"]').tooltip();
});
</script>
<script src="/static/js/upload.js"></script>
<script src="/static/js/dragAndDropUpload.js"></script>
<script src="/static/js/deleteImage.js"></script>
<script src="/static/js/setupImagesForCropping.js"></script
    >
<script src="/static/js/saveMammogramData.js"></script>

```

```

<script src="/static/js/deleteAll.js"></script>
<script src="/static/js/cropper.js"></script>
{%- endblock %}

```

## templates/predictResults.html

```

{% extends 'layout.html' %}

{% block head -%}
<link rel="stylesheet" href="/static/css/predictResults.css">
{%- endblock %}
{% block body %}
<div id="resultsContainer" class="container predictResults-
  container px-0">
  <div class="container-fluid">
    <h3 style="position:relative; top:80px;">RESULTS</h3>
  </div>
  <div id="resultsList" class="container-fluid predictResults-
    container mx-0 px-0" style="width: auto; background
    -color: MintCream; height:auto; position:relative; top:80
    px;">
    <ul id="PDFContent" class="list-group list-group-flush">
      {% for i in range(mammograms|length) %}
      <li class="list-group-item container">
        <div class="row">
          <div class="col-4">
            
          <div class="col">
            <h4>{{ mammograms[i].title }}</h4>
            <div class="row">
              <div class="col">
                <p>Patient name: {{ mammograms[i].patientName
                  }}</p>
                <p>Breast: {{ mammograms[i].breast }}</p>
                <p>Mammogram view: {{ mammograms[i].breastView
                  }}</p>
              </div>
              <div class="col">
                <p>Description: {{ mammograms[i].notes }}</p>
                <p>Verdict: {{ result[i] }}</p>
              </div>
            </div>
          </div>
        </div>
      </li>
      {% endfor %}
    </ul>
    <div class="container-fluid my-3" style="position:height:
      60px;">
      <a href="{{ url_for('pdfPredictResults') }}" target="
        _blank" class="btn btn-dark d-inline float-right">
        Generate PDF</a>
    </div>
  </div>
{%- endblock %}
{% block script -%}

```

```
{%- endblock %}
```

## templates/quiz.html

```

{% extends 'layout.html' %}

{% block head -%}
<link rel="stylesheet" href="/static/css/predictResults.css">
{%- endblock %}
{% block body %}
<div id="resultsContainer" class="container predictResults-
  container px-0">
  <div class="container-fluid">
    <h3 style="position:relative; top:80px;">EXAM</h3>
    <p style="position:relative; top:80px;">Multiple choice.
      Read the questions carefully, and choose the correct
      answer accordingly.</p>
  </div>
  <div id="resultsList" class="container-fluid predictResults-
    container mx-0 px-0" style="width: auto; background
    -color: MintCream; height:auto; position:relative; top:80
    px;">
    <ul class="list-group list-group-flush">
      <form id="exam" method="POST" action="{{ url_for('
        quizResults') }}">
        {% set count = [0] %}
        {% for quizItem in quizItems %}
        <li class="list-group-item container">
          <div class="row">
            {% if quizItem.questionImage != "" %}
            
            {% endif %}
            <div class="col">
              <div name="question-container container">
                <p>{{ count[0] + 1 }}. {{ quizItem.question }}</p>
              </div>
              <input type="hidden" id="idQuiz" name="id{{ count
                [0] }}" value="{{ quizItem.id }}">
              <div class="form-group">
                <div class="form-check">
                  <input class="form-check-input" type="radio"
                    name="answer{{ count[0] }}" id="question1"
                    value="{{ quizItem.otherChoices[0] }}">
                  <label class="form-check-label" for="question1
                    ">{{ quizItem.otherChoices[0] }}</label>
                </div>
                <div class="form-check">
                  <input class="form-check-input" type="radio"
                    name="answer{{ count[0] }}" id="question2"
                    value="{{ quizItem.otherChoices[1] }}">
                  <label class="form-check-label" for="question2
                    ">{{ quizItem.otherChoices[1] }}</label>
                </div>
                {% if quizItem.otherChoices[2] != "" %}
                <div class="form-check">

```

```

        <input class="form-check-input" type="radio"
            name="answer{{ count[0] }}" id="question3"
            value="{{ quizItem.otherChoices[2] }}">
        <label class="form-check-label" for="question3"
            ">{{ quizItem.otherChoices[2] }}</label>
    </div>
{% endif %}
{% if quizItem.otherChoices[3] != "" %}
    <div class="form-check">
        <input class="form-check-input" type="radio"
            name="answer{{ count[0] }}" id="question4"
            value="{{ quizItem.otherChoices[3] }}">
        <label class="form-check-label" for="question4"
            ">{{ quizItem.otherChoices[3] }}</label>
    </div>
{% endif %}
</div>
</div>
</li>
    {% if count.append(count.pop() + 1) %}{% endif %}
{% endfor %}
</form>
<div class="container-fluid my-3" style="position: absolute;
    height: 60px;">
    <button id="submitExam" type="button" class="btn btn
        -dark d-inline float-right">Submit</button>
</div>
</ul>
</div>
</div>
{%- endblock %}
{% block script -%}
    <script src="/static/js/submitQuiz.js"></script>
{%- endblock %}

```

## templates/quizResults.html

```

{% extends 'layout.html' %}

{% block head -%}
    <link rel="stylesheet" href="/static/css/predictResults.css">
{%- endblock %}

{% block body %}
    <div id="resultsContainer" class="container predictResults-
        container px-0">
        <div class="container-fluid">
            <h3 style="position: relative; top: 80px;">TOTAL SCORE:
                {{ finalScore }}</h3>
        </div>
        <div id="resultsList" class="container-fluid predictResults-
            container mx-0 px-0" style="width: auto; background
                -color: MintCream; height: auto; position: relative; top: 80
                px;">
            <ul class="list-group list-group-flush">
                {% set count = [0] %}
                {% for item in quizResultsData %}

```

```

<li class="list-group-item container">
    <div class="row">
        <div class="col-2 fluid-container">
            {% if item.quizResult == 'Correct' %}
                <i style="margin: auto;" class="fas fa-check-circle fa
                    -3x"></i>
            {% else %}
                <i style="margin: auto;" class="fas fa-times-circle fa
                    -3x"></i>
            {% endif %}
        </div>
        {% if item.questionImage != "" %}
            
        {% endif %}
    <div class="col">
        <div name="question-container container">
            <p>{{ count[0] + 1 }}. {{ item.questionRender }} </
                p>
        </div>
        <div class="form-group">
            {% set checked1 = 'disabled' %}
            {% set checked2 = 'disabled' %}
            {% set checked3 = 'disabled' %}
            {% set checked4 = 'disabled' %}
            {% if item.userAnswer == item.otherChoices[0] %}
                {% set checked1 = 'checked' %}
            {% elif item.userAnswer == item.otherChoices[1] %}
                {% set checked2 = 'checked' %}
            {% elif item.userAnswer == item.otherChoices[2] %}
                {% set checked3 = 'checked' %}
            {% else %}
                {% set checked4 = 'checked' %}
            {% endif %}
            <div class="form-check">
                <input class="form-check-input" type="radio"
                    name="answer{{ count[0] }}" id="question1"
                    value="{{ item.otherChoices[0] }}" {{ checked1
                        }}>
                <label class="form-check-label" for="question1"
                    ">{{ item.otherChoices[0] }}</label>
            </div>
            <div class="form-check">
                <input class="form-check-input" type="radio"
                    name="answer{{ count[0] }}" id="question2"
                    value="{{ item.otherChoices[1] }}" {{ checked2
                        }}>
                <label class="form-check-label" for="question2"
                    ">{{ item.otherChoices[1] }}</label>
            </div>
            <div class="form-check">
                <input class="form-check-input" type="radio"
                    name="answer{{ count[0] }}" id="question3"
                    value="{{ item.otherChoices[2] }}" {{ checked3
                        }}>
                <label class="form-check-label" for="question3"
                    ">{{ item.otherChoices[2] }}</label>

```

```

</div>
<div class="form-check">
  <input class="form-check-input" type="radio"
    name="answer{{ count[0] }}" id="question4"
    value="{{ item.otherChoices[3] }}" {{ checked4
  }}>
  <label class="form-check-label" for="question4
    ">{{ item.otherChoices[3] }}</label>
</div>
</div>
{% if item.quizResult != 'Correct' %}
<p>The correct answer is {{ item.correctAnswer }}.</
  p>
  {% endif %}
</div>
</div>
</li>
  {% if count.append(count.pop() + 1) %}{% endif %}
  {% endfor %}
</ul>
</div>
</div>
{%- endblock %}
{% block script -%}
  {%- endblock %}

```



## **XI. Acknowledgement**

I would like to thank my thesis adviser, Dr. Vincent Peter C. Magboo, for the guidance on my SP, and for defending me even when the whole panel is against me. Thank you, sir! I would also like to mention Edward Francis Lacanlale for teaching me everything I need to know regarding the all-too-complex caffe, assisting me when I was in dire need, and for giving me hope when my laptop ran into a problem and couldn't boot anymore. You are the real MVP.

A special mention goes out as well to my clutch buddies, John Bengemin Uy and Michael Ramos, for the camaraderie giving me assurance that I won't be going alone on this one.

To my dear friends and blockmates, it was a pleasure.