# DEVELOPING A CONFIDENCE MEASURE BASED EVALUATION METRIC FOR BREAST CANCER SCREENING USING BAYESIAN NEURAL NETWORKS

by

Anika Tabassum, BA Computer Science, McGill University 2013

A Major Research Project
presented to Ryerson University
in partial fulfillment of the requirements for the degree of

Master of Science
in the Program of
Data Science and Analytics

Toronto, Ontario, Canada, 2019

# AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A MAJOR RESEARCH PAPER (MRP)

I hereby declare that I am the sole author of this Major Research Paper. This is a true copy of the MRP, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this MRP to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this MRP by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my MRP may be made electronically available to the public.

Anika Tabassum

# DEVELOPING A CONFIDENCE MEASURE BASED EVALUATION METRIC FOR BREAST CANCER SCREENING USING BAYESIAN NEURAL NETWORKS

Anika Tabassum

Master of Science 2019

Data Science and Analytics

Ryerson University

## ABSTRACT

Screening mammograms is the gold standard for detecting breast cancer early. While a good amount of work has been performed on mammography image classification and many of the recent ones have made use of deep neural networks successfully, there has not been much exploration into the confidence or uncertainty measurement of the classification, especially with Bayesian neural networks. In this paper, we propose a new evaluation criterion based on confidence measurement for breast cancer mammography image classification, so that in addition to classification accuracy, it provides a few numeric parameters that can be tuned to adjust the confidence level of the classification. We demonstrate the use of Bayesian neural networks and transfer learning in the process of achieving that. We also demonstrate the expected behaviour resulting from tuning of the parameters and conclude by saying that the approach is extendable to any domain in general and any number of classes.

***Key words: Bayesian Neural Networks, Transfer Learning, Deep Learning, Breast Cancer Screening, Confidence Measurement, Uncertainty Measurement, Mammography***

# ACKNOWLEDGEMENTS

I am thankful to: my mother, for her unconditional love and support, and for solving all of my problems; my father and brother, for their unconditional love**;** Dr Naimul Khan, for providing valuable guidance and supervision for my Major Research Project; my husband, for being cute and helpful.

# TABLE OF CONTENTS

AUTHOR'S DECLARATION ..... ii
ABSTRACT ..... iii
ACKNOWLEDGEMENTS ..... iv
List of Figures ..... vi
List of Tables ..... vii
1. Introduction ..... 1
2. Literature Review ..... 3
3. Exploratory Data Analysis ..... 5
    3.1. Dataset Description ..... 6
4. Data Preprocessing ..... 8
    4.1. Image Resizing ..... 8
    4.2. Normalization ..... 8
    4.3. Augmentation ..... 8
    4.4. Training-Test Split ..... 9
    4.5. Data Preparation Scheme ..... 9
5. Deterministic Training and Evaluation ..... 10
    5.1. Network Architecture – Adoption of ResNet-18 ..... 11
    5.2. Training the Deterministic Network ..... 12
    5.3. Evaluation and Results ..... 12
6. Bayesian Training and Evaluation ..... 13
    6.1. Conceptual Setup ..... 13
    6.2. Network Architecture ..... 15
    6.3. Training ..... 17
    6.4. Evaluation and Results ..... 17
7. Discussion and Future Work ..... 18
8. References ..... 23
9. Appendix ..... 25

# LIST OF FIGURES

Figure 1 – Class Distribution ..... 7
Figure 2 – Sample benign and malignant image patches ..... 7
Figure 3 – Original ResNet-18 and our adopted architectures ..... 11
Figure 4 – Bayesian Neural Network Architecture ..... 16
Figure 5 – Accuracy and Coverage Trends Against N and P ..... 20

# 1. LIST OF TABLES

Table 1 – Accuracy and Coverage Values with N and P tuning ..... 19

# 1. INTRODUCTION

Breast cancer is the most common cancer in women around the world according to the World Health Organization [14]. The key to breast cancer control is the early detection to improve breast cancer outcome and survival [14]. Mammography is the most common screening technology for breast cancer. It is a type of imaging that uses a low- dose X-ray system to examine the breast and is the most reliable method for screening breast abnormalities [15] before they become clinically perceptible. Screening mammography is done for detecting breast cancer. However, one big challenge here is low contrast in the mammogram images, which makes it hard for radiologists to interpret the results [1]. Therefore, the use of computer aided diagnosis (CAD) has been on the rise for breast cancer screening [15].

Computer aided detection is a pattern recognition process that helps radiologists detect potential abnormalities such as calcifications, masses, and architectural distortions [16]. To accomplish this, there have been traditional approaches based on heavy feature engineering, as well as recent approaches based on deep convolutional neural networks. However, for a crucial task like cancer image screening, just classifying an image to a particular class (like benign or malignant) is not really enough, because it lacks any confidence or uncertainty measure associated with the classification [18]. For example, if an image is classified as malignant, the radiologist might be interested in knowing how confident the CAD system actually is that it is malignant. In others, the classification accuracy on known images is not a sufficient evaluation criterion, it has to be accompanied by a confidence or uncertainty measure. This is an aspect where there has not been too much exploration yet, and we chose to do something in this area for my MRP.

While deep convolutional neural networks have been excellent in most image classification tasks, they output a single probability for each class, and due to a softmax function at the output layer, they squish one class output probability score and maximize the other(s), leading to an overconfident decision for one class [17]. This is one major problem with a deterministic point-estimate neural network, where each of the weights is just a single point. While this limitation can be greatly tackled via using regularization techniques [17], that still does not provide any confidence or uncertainty measurement for the classification, and this is where Bayesian neural networks can come into play. In addition to being robust to overfitting, Bayesian neural networks offer uncertainty/confidence estimates via its parameters in the form of posterior probability distributions. By using a prior probability distribution to integrate out the parameters, the average is computed across many models during training, which prevents overfitting. In this project, our goal, however, is not to explore the internals of Bayesian approaches but to apply Bayesian deep learning to breast cancer image classification and come up with a new evaluation criterion based on confidence/uncertainty measure.

We obtain the mammography images from CBIS-DDSM [1] and the classification task is essentially binary (benign vs malignant). We first train a deterministic point-estimate neural network using the pretrained ResNet-18 architecture with some modifications, thus leverage transfer learning. Then for saving computational resources we separate the feature extractor from this deterministic network to generate lower dimensional features and feed those to a separate smaller network which acts as our Bayesian neural network. Having computed the posterior distributions by applying Stochastic Variational Inference (SVI) [9], we introduce two tunable parameters **N (**sampled network fraction**)** and **P (**minimum probability**)** which together (both are

explained in detail in section 6) can be used as a confidence measure and be tuned to adjust the confidence level. We also demonstrate that higher confidence results in lower coverage of the classification, i.e., some images are denied any class due to lack of confidence. We propose that the tuple ***(accuracy, coverage, N, P)*** can be our new evaluation criterion where ***(N, P)*** is the confidence measure, and the overall approach can in general be applicable to any domain beyond medical imaging and any number of classes.

The rest of this report is organized as follows: section 2 discusses related literature, section 3 discusses some exploratory analysis on the data, section 4 discusses the preprocessing we needed to perform on the data, section 5 discusses the deterministic portion of our training process and section 6 discusses the Bayesian training process. Finally, in section 7 we conclude by throwing some insight on possible future work.

## 2. LITERATURE REVIEW

The purpose of this project is to eventually come up with a confidence measure based evaluation criterion for breast cancer image classification, using the application of Bayesian deep learning. While there has been quite a bit of work performed out there on application of deep learning for medical image classification, not too many specifically explore the Bayesian option.

*Tsochatzidis et al.* [10] performed a comparative study on applying CNNs for breast cancer image diagnosis. They made use of Alexnet, VGG, GoogLeNet, Inception Networks and ResNet and evaluated based on ROC curve (AUC) and classification accuracy. They showed that under fine-tuning scenario, pretrained networks achieve superior performance over networks trained

from scratch. As we will see later in this report, we also make use of transfer learning using pretrained networks.

*Agarwal et al.* [11] showed a framework for automatic mass detection using deep CNNs. They used a patch-based CNN method for automated mass detection; investigated use of transfer learning CNN is trained using CBIS-DDSM [1], then transferred and tested onto a smaller database called INbreast. The framework is initialized by extracting small regions of the image (patches) for training, obtaining unseen testing patches as mass and non-mass, recombining patches to reconstruct the mammogram and finally using the classification probabilities to obtain mass probability map and obtain probable mass region. They also tried with VGG16, ResNet50 and InceptionV3 and found that the framework earns a higher accuracy using transfer learning (pretrained weight initialization) than it does using random weight initialization.

*Xi et al.* [2] performed binary classification of mammography images using transfer learning. They modified pretrained CNNs at output layers to have 2 output classes. Output layers were fine-tuned while the first part of the network is frozen. Fine-tuned neural network is then used to localize mammographic abnormalities in full-size mammograms. They made use of AlexNet, VGGNet, GoogleLeNet and ResNet and showed that VGGNet achieves the best overall accuracy while ResNet performs best for computing class activation maps. We will see later in this report (section 5) that the first part of our training (the deterministic training) largely tries to mimic their work in terms of data preparation and network architecture selection.

*Rampun et al.* [12] performed classification of mammographic microcalcification clusters with confidence levels. They studied distribution of classifiers' probability outputs and uses it as an additional confidence level metric to indicate reliability, applying a whole bunch of classifiers

including logistic regression, multilayer perceptron, random forest, C4.5, k-nearest neighbours, Adaboost, Logistic Model Trees, Bayesian networks and SVM. The big question the asked is: *if a case is classified malignant, what is the degree of certainty that it is actually malignant?* They found that in terms of confidence levels the kNN and LMT classifiers have the highest true positive and true negative scores. They conclude that in breast CAD systems, the CA or AUC metric alone does not provide a complete representation of reliability. Although they do not make use of Bayesian neural networks, our project is greatly motivated by this work given that we are also looking for a confidence measure based evaluation criteria for classification.

Although not particularly in the domain of medical imaging, *Harper and Southern* [13] showed a Bayesian deep learning framework for prediction emotion from heartbeat by introducing a tunable confidence measure. Their confidence measure is based on the percentage of the output distribution that lie within a given class zone. Our work is also partially motivated by this work and we will see later in section 6 that we also introduce a few tunable parameters for confidence measure.

Having reviewed the mentioned work above and being motivated by some of them, we have decided to leverage Bayesian neural networks along with transfer learning to come up with an evaluation criterion based on a confidence measure controlled by a few tunable parameters. Next, we dive into the core discussion of our work starting with exploratory analysis.

## 3. EXPLORATORY DATA ANALYSIS

Our dataset consists solely of image data. The data has been obtained from the CBIS-DDSM

(Curated Breast Imaging Subset of DDSM) [1] dataset. It is an updated and standardized version of the Digital Database for Screening Mammography (DDSM). The original DDSM database consists of 2,620 scanned filmed mammography studies, whereas the CBIS-DDSM includes a subset of the DDSM data curated by a trained mammographer.

The CBIS-DDSM dataset contains a few sub datasets. It includes full mass images and ROI (Region of Interest) cropped mass images, as well as full calcification images and ROI-cropped calcification images. Since our underlying task is primarily a classification one (benign vs malignant), we use the mass images (as opposed to calcification) for our purpose. However, computer-aided mammography is a challenging problem and cannot be treated as a simple image classification task. The reason is that abnormalities within a whole image are located in small regions. For example, a typical full mammogram with a resolution of 3000x4600 (width and height in pixels) contains an abnormality region of size only about 200x200 (pixels). For this reason, instead of the full mass images, we resort to using the cropped ROI image patches.

## 3.1 Dataset Description

The ROI-cropped dataset contains 1696 labelled grayscale images. Of those, 912 are labelled as benign and 784 are labelled as malignant. So, the dataset is reasonably class-balanced. **Figure 1** shows the distribution percentage of the benign and malignant classes in the dataset. The images provided are in DICOM format and not constant in sizes (width and height).

**Figure 2** shows samples of one benign and one malignant image patch. Labels are provided in a

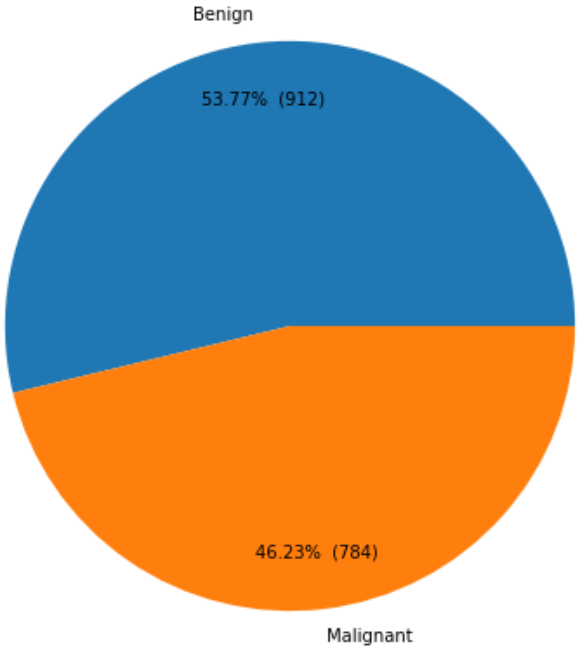
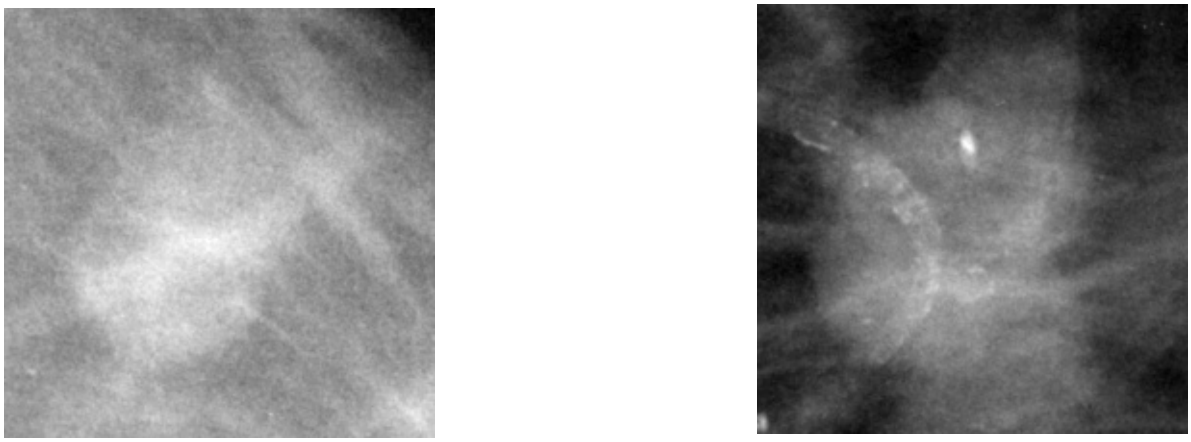



**Figure 1: Class Distribution**



(a) Benign (b) Malignant

**Figure 2: Sample benign and malignant image patches**

separate CSV file along with some other meta information, and we had to match the labels with the images based on their directory paths. The dataset is pre-divided into a training and a test set, with the training set containing 1318 image patches and the test set containing 378 image

patches., although as we will see later, we don't adhere to this pre-division and create our own training and test datasets. Given that it is mainly pre-collected and pre-labelled image data we are dealing with, there is not too much analytical exploratory analysis to be done here, aside from the simple dataset statistics mentioned above. However, there is some interesting pre-processing that can be done which we discuss in the next section.

# 4. DATA PREPROCESSING

## 4.1 Image Resizing

Since we will be using deep neural networks for our classification and given that most recent deep neural networks use inputs of fixed sizes (in particular the size 224 x 244 pixels), the first bit of preprocessing we perform is to resize all the images to 224 x 224 pixels.

## 4.2 Normalization

To make sure the distribution of the pixel values lies with the same range (-1 to 1), we perform standard normalization of all the images with a mean of 0.5 and standard deviation of 0.5 as well.

## 4.3 Augmentation

Given that our dataset size is not really that big, there is ample chance that it will be prone to overfitting. To reduce the chance of overfitting, we perform data augmentation by applying random rotation (from 0 to 360 degrees) and random vertical and horizontal flips on the images. This is applied during training only. The idea is that in each epoch, a random combination of augmentation will be applied to every image and so after a sufficient number of epochs, we will

have covered a reasonably large image set, even though the original (not augmented) dataset might not be that big.

## 4.4 Training-Test Split

As mentioned before, although the CBIS-DDSM comes with a pre-division of the dataset into a training and a test portion, for more generalization, we merge these two datasets and then split again into our own training and test datasets. This splitting is done 80:20 training to test ratio and in a stratified manner with regards to the class (benign vs malignant) distribution. After this split, we have **1357** training image and **339** test images. We create 5 such stratified splits so that each such split can be used as a validation set cross-validation purpose to test the generalization of our approach.

## 4.5 Data Preparation Scheme

We found that with preprocessing like resizing, normalization, augmentation etc, it is quite time-consuming and not so performant to let the training process read images off the disk. That way, a significant portion of the training time is taken by the disk image read and preprocessing steps. To get rid of this problem, we have decided to use an HDF5 file as a source of all our data, since HDF5 format allows the managing of large amount of numeric data very efficiently. We perform the resizing of all the images (train and test), convert them to numeric arrays and then store them and the associated labels in the HDF5 file. We also store the cross-validation indices of our images in the same HDF5 file. Once we have all the information stored in the HDF5 file, it becomes dramatically faster for the training and evaluation processes to run.

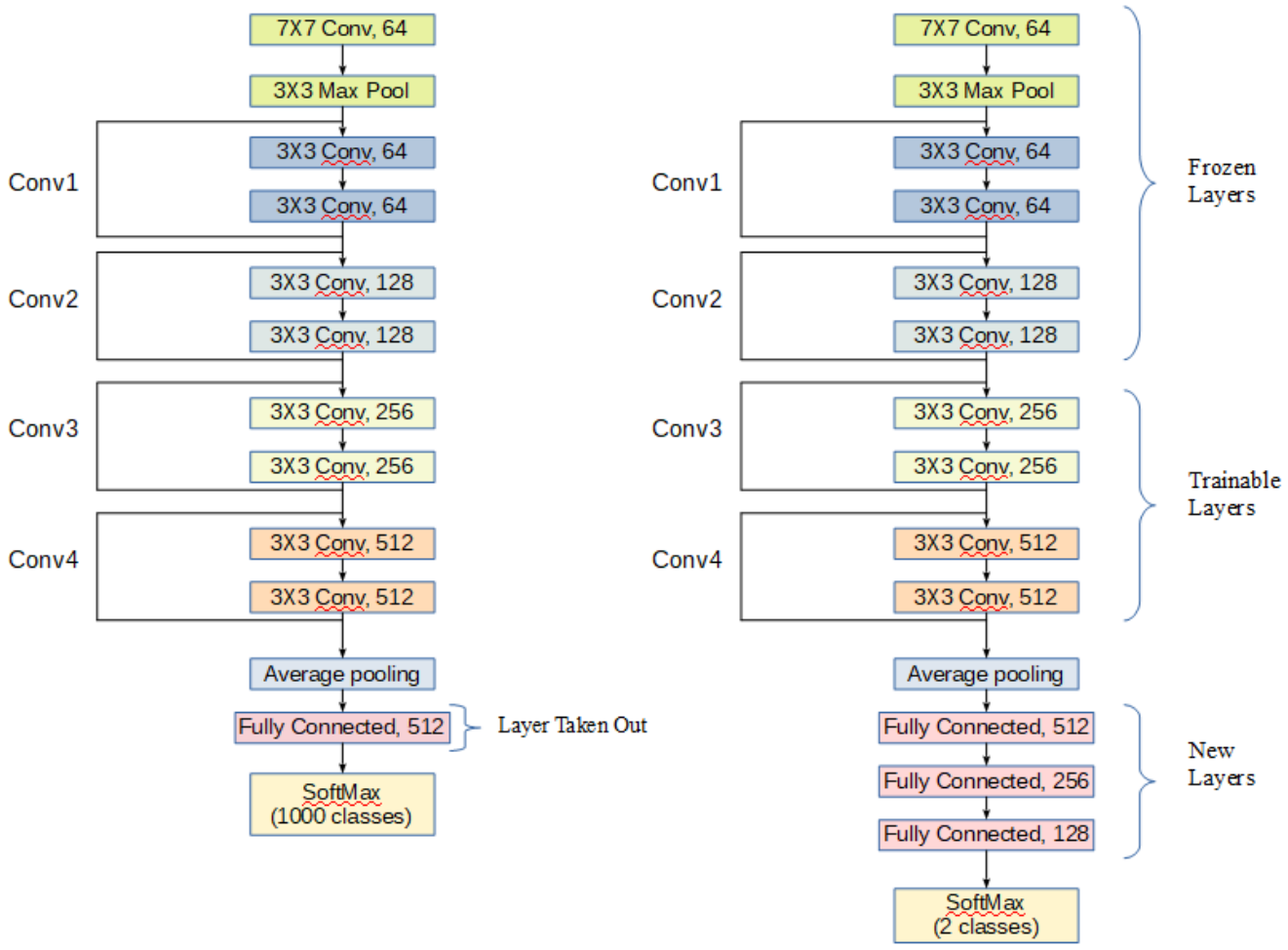
Having preprocessed and prepared our data, we now discuss the training and evaluation, which consists of 2 stages – deterministic training/evaluation and Bayesian training/evaluation.

# 5. DETERMINISTIC TRAINING AND EVALUATION

Our goal is to leverage Bayesian neural networks to come up with a new evaluation metric. However, to achieve that, we first need to make sure we have a good architecture of a deterministic neural network, because the Bayesian neural network will be based on the deterministic one. Given that there are already well-studied and researched deep neural network architectures out there, we have decided to try out some of these to see which one performs best for our case. In particular we have tried out AlexNet [3], VGGNet [4] and ResNet [5]. As for VGGNet and ResNet, we have tried out the VGG-16 and ResNet-18 variations. Since there are pretrained versions (trained on the large ImageNet database [6]) of these networks available, we adopted transfer learning by leveraging these pretrained networks.

However, since the ImageNet database has 1000 categories of images covering a lot of domains and ours has only 2 categories (benign and malignant) and even that only in the medical mammography domain, just using the pretrained networks as is and reducing the category size to 2 is not a good idea. Therefore, we try to mimic the approach in [2] and in each case (AlexNet, VGG-16 and ResNet-18), we drop the last fully connected layer and replac that with our own block of trainable fully connected layers. Also, we make a few later convolutional blocks trainable while freezing the earlier part of the networks. This setting allows us to learn domain specific features on top of the pretrained networks. In our case, ResNet-18 turned out to be the best in terms of learning and performance generalization. For this reason, for the rest of this

section and also the report, we will confine the discussion to the results obtained by ResNet-18 only.




(a) Original Resnet-18 (b) Our Architecture

**Figure 3: Original ResNet-18 and our adopted architectures**

## 5.1 Network Architecture - Adoption of Resnet-18

**Figure 3** shows the original ResNet-18 architecture (a) and our adopted architecture (b). As can be seen, the earlier portion of the original network is mainly convolutional which acts as a feature generator, while the later part is mainly a fully connected layer followed by a softmax,

which is the classifier. There are 4 residual convolutional blocks **Conv1, Conv2, Conv3** and **Conv4.** In our adopted architecture, we drop the fully connected layer and replace it by 3 new fully connected layers as shown. Also, of the four residual convolutional blocks, we make the latter 2, namely **Conv3** and **Conv4** as trainable, meaning that even though the network is pretrained, when we train with our mammography images, these layers will be further trained. We freeze the earlier part of the network including **Conv1** and **Conv2.** We also modify the softmax layer to have only 2 classes as per our need. This setting gives us a way of leveraging transfer learning, as well as incorporating domain specific learning on top of it.

## 5.2 Training the Deterministic Network

Now that we have our network architecture set, we train it via our **1357** training images which are **224x224** in size. We train for 100 epochs and apply the random augmentation discussed in the previous section on each image. We use a batch size of **8** and an initial learning rate of **0.0001** for the earlier part of the network (except for the fully connected part). For the fully connected part we gradually increase the initial learning rate layer by layer, **0.0001** for the first layer, **0.001** for the second one and **0.002** for the last one. We also use an exponential learning rate decay scheme and an **Adam** optimizer. The training is done using the **PyTorch** [7] framework on **Google Colab** platform which uses a **Nvidia Tesla K80 GPU** and took approximately **12 minutes** to finish.

## 5.3 Evaluation and Results

During training, we keep track of the best checkpoint based on the performance on the validation set. After training in this scheme, we achieve **81%** classification accuracy on the test set on the

average. This is a reasonable accuracy and at this point, we start planning on the Bayesian training.

# 6. BAYESIAN TRAINING AND EVALUATION

The deterministic neural network gives us an accuracy value but doesn't give us a confidence or uncertainty measure associated with that value. To get a confidence measure, our next step is coming up with a plan for Bayesian training.

## 6.1 Conceptual Setup

To get a confidence measure, our plan is:

- Train a neural network by initializing the weights and biases with random priors (like a Gaussian distribution with 0 mean and unit standard deviation) and then train the network by applying Stochastic Variational Inference (SVI) [9] to get the posterior distribution.
- Once we have the posterior distributions, sample **S** number of networks from it. Notice that each of these **S** networks is a deterministic network by itself. Conceptually speaking, the higher the value of **S,** the better.
- Classify each image by each of the **S** networks. Record the probabilities for both classes (benign and malignant) for each image.
- We introduce two parameters **N** and **P,** where **N** is a fraction percentage of **S,** denoting the fraction of the **S** number of networks that have a minimum probability **P** on a certain image being of a particular class (either benign or malignant).

- Now that we have two tunable parameters **N** and **P,** we can define a confidence measure using these 2. For example, if **S = 1000**, then **N = 0.6** and **P = 0.7** would mean at least **600** networks out of the **1000** have to have a probability of at least **0.7** for an image being either benign or malignant (but not a mix-up of both)**,** otherwise the image would be skipped for classification. In other words, by incorporating both **N** and **P** in the confidence measure, we account for agreement among a portion of the sampled networks and also how strongly each network feels about the classification.

Under the above settings, the natural expectation would be that as N and P go higher (higher confidence, lower uncertainty), we should be getting higher accuracy values while as N and P go lower (higher confidence, lower uncertainty), the accuracy should decrease. However, notice that raising the value of N and P might also result in some images being skipped for classification. For example, consider a case where **S = 1000, N = 0.9** and **P = 0.9**, which is demanding that at least **900** out the **1000** networks have to have a probability of at least **0.9** for an image being either benign or malignant – this might result in a lot of images being skipped for classification, since we are demanding too high a confidence. This is the case of lower coverage. At higher values of **N** and **P** (higher confidence), we will have lower coverage (many images skipped), but the accuracy on the covered images will be high. On the other hand, at lower values of **N** and **P** (lower confidence), we will have higher coverage (not too many images skipped) but the accuracy on the covered images would be lower. In short, tuning the values of **N** and **P,** gives us a way to decide where in the accuracy-coverage tradeoff we want to settle. Therefore the **N** and **P** values, along with the accuracy and coverage, comprise our new evaluation metric.

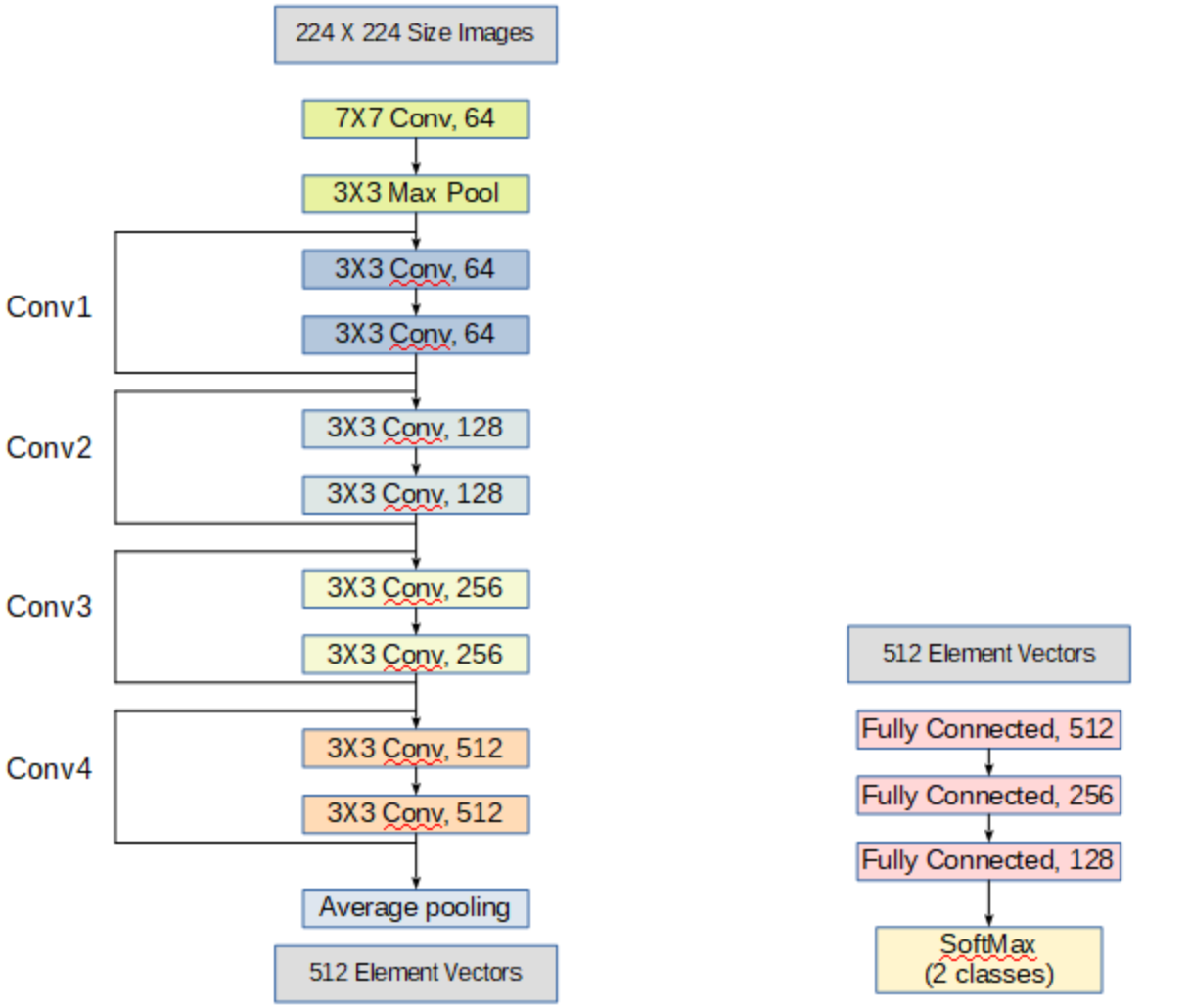
Now that we have our conceptual plan laid out, the next challenge is applying this plan successfully on top of the deterministic network we trained in section 5.

## 6.2 Network Architecture

One straightforward approach could have been just to use the deterministic neural network architecture we have and initialize the weights and biases with random priors (like a Gaussian distribution with 0 mean and unit standard deviation) and then train the network using SVI. However, there are a few empirical problems associated with this approach:

- Our adopted architecture from ResNet-18 has over 11 million parameters, so training a distribution for each of these parameters will be very time consuming and impractical.
- There does not seem to be enough tool support and advancement out there for applying Bayesian learning on top of transfer learning

Keeping these in mind, we opt for a slightly different sort of network architecture for the Bayesian training. The new architecture is shown in **Figure 4**. Having trained our deterministic neural network as described in section 5, we extract the convolutional portion out of that as our **feature generator**. We create a separate network consisting of just 3 fully connected layers to be used as our Bayesian neural network. This separate network has the exact same layout as the fully connected part in the original deterministic network. We will initialize the weights of this network with normally distributed priors and infer the posteriors using SVI. This network only has just over 164 thousand parameters so it will not be computationally expensive.




(a) Feature Generator (already trained) (b) Bayesian Neural Network

**Figure 4: Bayesian Neural Network Architecture**

The idea for training is:

- We will use the feature generator to generate 512-element feature vectors from our 224x224 images. Notice that there is no training involved here since the feature generator was already trained during the deterministic neural network training phase in section 5. We will do this for both training images and test images.

- We will feed these 512-element vectors (only the training ones) to the relatively smaller Bayesian neural and infer the posteriors using SVI.
- Once this is done, we will use the 512-element test features along with the concept laid out in section 6.1 to evaluate and tune confidence measurements.

Notice that with this hybrid approach, we are able to retain the benefits of transfer learning, as well as apply Bayesian learning on top of that within an acceptable time bound.

## 6.3 Training

We first generate the 512- element features using the feature generator. Once again, we apply random augmentation as described in section 4.3 in a way that we have 10 augmented images for every original training image (no augmentation is done for test images). Since we have **1357** training images, we effectively end up generating **13570** training images and correspondingly **13570** number of 512-element vectors from the feature generator. We apply SVI using **Pyro** software package [8]. The hyper-parameters we use (optimizer, learning rate) are the same as what we used during the deterministic training, and we let it run for 10 epochs on the same Google Colab environment using a **Nvidia Tesla K80 GPU**. The process took approximately 30 minutes to finish.

## 6.4 Evaluation and Results

Following the conceptual plan laid out in section 6.1, we sample 1000 networks (**S = 1000**) from the posterior distributions learnt via training and perform evaluation by tuning the values of **N** and **P**. We observe the following interesting findings during evaluation:

- The first bit of evaluation we perform is what we call forced prediction. In this case, we ignore **N and P** altogether and just force our Bayesian neural network to predict a class for every image. The way we do it is by taking the average predicted probability for both classes (benign and malignant) for each image and taking the class that has the higher average as the predicted class. This gives us **100%** coverage and a prediction accuracy of **81**%. Note that this is exactly the same accuracy we got using the single deterministic network in section 5.
- Now we bring in **N** and **P** in the scene and this is where things start looking really interesting. Bear in mind that we have **339** test images. We find that when we set N above 0.9, all test images are skipped and the Bayesian network has no coverage regardless of the value of P, which is kind of expected because it's unlikely that 95% of the networks would have a minimum probability for a class for any image. In general, if we keep N to a moderate value like 0.5 and then vary P, we notice an upward trend in the accuracy and a downward trend in the coverage, which is expected. An almost similar trend shows up if we hold P at 0.5 and vary N instead, which is also expected. The exact figures have been shown in **Table 1** and the trends have been shown in **Figure 5.**

Based on the observations, the conceptual expectation we laid out in section 6.1 is verified. Thus, instead of only accuracy, we can formulate a tuple like ***(accuracy, coverage, N, P)*** which could be our new evaluation criterion with ***(N, P)*** being the measure of confidence.

# 7. DISCUSSION AND FUTURE WORK

The takeaway from the findings in the last section is that the parameters **N** and **P** can be tuned to

| N | P | Total Images | Skipped Images | Accuracy | Coverage |
|---|---|---|---|---|---|
| 0.95 | 0.5 | 339 | 339 | NA | 0 |
| 0.9 | 0.5 | 339 | 337 | 0.5 | 0.0059 |
| 0.85 | 0.5 | 339 | 311 | 0.964286 | 0.082596 |
| 0.8 | 0.5 | 339 | 285 | 0.944444 | 0.159292 |
| 0.75 | 0.5 | 339 | 257 | 0.939024 | 0.241888 |
| 0.7 | 0.5 | 339 | 177 | 0.925926 | 0.477876 |
| 0.65 | 0.5 | 339 | 124 | 0.897674 | 0.634218 |
| 0.6 | 0.5 | 339 | 70 | 0.884758 | 0.79351 |
| 0.55 | 0.5 | 339 | 31 | 0.847403 | 0.908555 |
| 0.5 | 0.5 | 339 | 2 | 0.818991 | 0.9941 |
| 0.45 | 0.5 | 339 | 0 | 0.817109 | 1 |
| 0.4 | 0.5 | 339 | 0 | 0.817109 | 1 |
| 0.35 | 0.5 | 339 | 0 | 0.817109 | 1 |
| 0.3 | 0.5 | 339 | 0 | 0.817109 | 1 |
| 0.25 | 0.5 | 339 | 0 | 0.817109 | 1 |
| 0.2 | 0.5 | 339 | 0 | 0.817109 | 1 |

(a) Accuracy and Coverage for different values of N (P = 0.5)

| N | P | Total Images | Skipped Images | Accuracy | Coverage |
|---|---|---|---|---|---|
| 0.5 | 0.95 | 339 | 85 | 0.877953 | 0.749263 |
| 0.5 | 0.9 | 339 | 71 | 0.876866 | 0.79056 |
| 0.5 | 0.85 | 339 | 55 | 0.862676 | 0.837758 |
| 0.5 | 0.8 | 339 | 44 | 0.850847 | 0.870206 |
| 0.5 | 0.75 | 339 | 35 | 0.845395 | 0.896755 |
| 0.5 | 0.7 | 339 | 31 | 0.844156 | 0.908555 |
| 0.5 | 0.65 | 339 | 25 | 0.83758 | 0.926254 |
| 0.5 | 0.6 | 339 | 16 | 0.826625 | 0.952802 |
| 0.5 | 0.55 | 339 | 10 | 0.817629 | 0.970501 |
| 0.5 | 0.5 | 339 | 2 | 0.818991 | 0.9941 |
| 0.5 | 0.45 | 339 | 0 | 0.817109 | 1 |
| 0.5 | 0.4 | 339 | 0 | 0.811209 | 1 |
| 0.5 | 0.35 | 339 | 0 | 0.80826 | 1 |
| 0.5 | 0.3 | 339 | 0 | 0.80826 | 1 |
| 0.5 | 0.25 | 339 | 0 | 0.80826 | 1 |
| 0.5 | 0.2 | 339 | 0 | 0.80826 | 1 |

(b) Accuracy and Coverage for Different values of P (N = 0.5)

**Table 1: Accuracy and Coverage Values with N and P tuning**

a desirable level and the higher the values of these parameters will be, the higher confidence we will have in the predictions/classifications, in exchange for possibly lower coverage. For a domain like medical mammography image classification, it will be up to the mammography

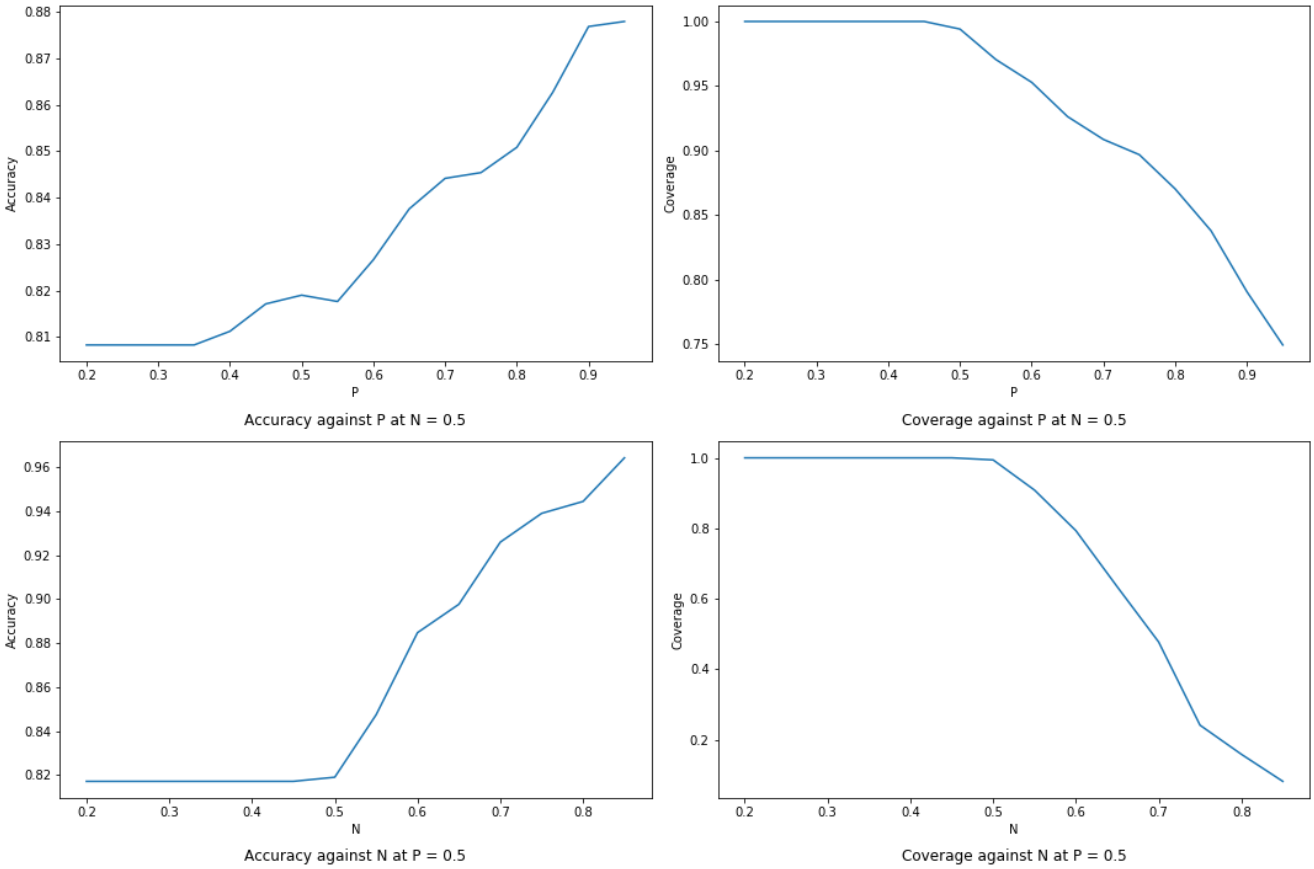



**Figure 5: Accuracy and Coverage Trends Against N and P**

experts as to what value of **N** and **P** could be reasonable. After setting reasonable values for N and P, previously unseen images that will be classified by the Bayesian network (either as benign or malignant) are more likely to be correctly classified, whereas images that will be skipped and denied classification would demand further investigation. Notice that the overall approach described in sections 5 and 6 can in general be applied to any domain (beyond mammography) and any number of classes, and thus can be used as some sort of skeleton framework for an uncertainty/confidence measurement task.

A lot of scope for future work remains. We will discuss some of it before concluding. Although not explicitly discussed in this report, during training of our deterministic and Bayesian neural

networks, we noticed that the more accurate the deterministic network is, the more consistent the behaviour of the Bayesian network is also. This means that without a reasonably good feature extractor that we used as a precursor for the Bayesian training, the expected accuracy-coverage trade-off behaviour is not quite observed. It was only after we could achieve over 80% accuracy through the deterministic neural network (and hence the feature generator) that the expected upward and downward trends in **Figure 5** were achieved via the Bayesian neural network. This implies that before Bayesian approach can be relied upon, we first need a good network architecture to be trained with proper hyper parameter tuning. In our case, that was the ResNet-18 with our custom adoption. However, more complex network architectures can be explored including some in the ResNet family (like ResNet-50, ResNet-152 etc). Also, during transfer learning, we noticed that the fewer layers we freeze and the more layers we make trainable, the accuracy generally increases. With a more complex network architecture, tuning how many layers to freeze would be one interesting thing to explore.

Our Bayesian neural network architecture was a relatively simple one with just 3 fully connected layers. However, a more complex one can be explored in the future. Also, the prior distributions we assigned to the Bayesian layer parameters were just simple Gaussian distributions with zero mean and unit standard deviation. Ideally though, these priors should come from a more rigorous source including domain knowledge.

Due to the apparent shortage of proper tooling in combining transfer learning with Bayesian learning, we could not make the Bayesian learning process end to end with just one single network architecture and instead broke it down to the deterministic feature generator and

Bayesian fully connected network. However, an end to end Bayesian learning approach on top of transfer learning is definitely something to be explored in the future. In that case, the priors to be assigned to the Bayesian layer parameters might as well come from the parameter values learnt for deterministic training.

To conclude, we have shown the potential of Bayesian approaches to be able to propose a new evaluation metric based on confidence measure and also laid out a few possible options for future work.

# 8. REFERENCES


[1] ***Cancer Imaging Archive***. https://wiki.cancerimagingarchive.net/display/Public/CBIS-DDSM

[2] P. Xi, C. Shu and R. Goubran. ***Abnormality Detection in mammography using Deep Convolutional Neural Networks***. 2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA).

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton**.** ***ImageNet Classification with Deep Convolutional Neural Networks***.NIPs 2012, Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Pages 1097-1105.

[4] K. Simonyan and A. Zisserman. ***Very Deep Convolutional Networks for Large-Scale Image Recognition***. https://arxiv.org/abs/1409.1556

[5] K. He, X. Zhang, S. Ren and J. Sun. ***Deep Residual Learning for Image Recognition***. https://arxiv.org/abs/1512.03385

[6] **ImageNet**. http://www.image-net.org/

[7] **PyTorch**. https://pytorch.org/

[8] **Pyro**. https://pyro.ai/

[9] M. D. Hoffman, D. M. Blei, C. Wang and J. Paisley. ***Stochastic Variational Inference***. Journal of Machine Learning Research, 2013. Volume 14, pages 1303 – 1347.

[10] L. Tsochatzidis, L. Costaridou and I. Pratikakis. ***Deep Learning for Breast Cancer Diagnosis from Mammograms - A Comparative Study.*** Journal of Imaging, 2019, 5(3), 37.

[11] R. Agarwal, O. Diaz, X. Llado, M. H. Yap and R. Marti. ***Automatic Mass Detection in Mammograms using deep convolutional neural networks.*** Journal of Medical Imaging, 2019, 6(3), 031409.

[12] A. Rampun, H. Wang, B. Scotney and P. J. Morrow. ***Classification of Mammographic Microcalcification Clusters with Machine Learning Confidence Levels.*** 14th International Workshop on Breast Imaging, July 2018.

[13] R. Harper and J. Southern. ***A Bayesian Deep Learning Framework for End-to-End Prediction of Emotion from Heartbeat***. https://arxiv.org/abs/1902.03043

[14] World Health Organizartion. ***Breast Cancer: Prevention and Control***. https://www.who.int/cancer/detection/breastcancer/en/

[15] J. Tang, RM Rangayyan, j. Xu, I. E. Naqa and Y. Tang**. *Computer-aided detection and diagnosis of breast cancer with mammography: recent advances*.** IEEE Transactions on Information Technology in Biomedicine, 2009 Mar;13(2):236-51.

[16] R. A. Castellino**. Computer aided detection (CAD): an overview.** *Cancer Imaging*, vol. 5, no. 1, pp. 17–19, 2005.

[17] K. Sridhar. ***Bayesian neural network series***. https://medium.com/neuralspace/bayesian-neural-network-series-post-1-need-for-bayesian-networks-e209e66b70b2

[18] B. M. Geller, A. Bogart, P. A. Carney, J. G. Elmore, B. S. Monsees and D. L. Miglioretti. ***Is Confidence of Mammographic Assessment a Good Predictor of Accuracy?*** AJR Am J Roentgenol. 2012 Jul; 199(1): W134–W141.

# 9. APPENDIX

**Github project link:** https://github.com/atabas/Major-Research-Project

**Code for HDF5 File Generation**: https://github.com/atabas/Major-Research-Project/blob/master/hdf5_generation.ipynb

**Code for Deterministic and Bayesian Training and Evaluation**: https://github.com/atabas/Major-Research-Project/blob/master/ddsm_experiment_cropped_cv.ipynb

**Result CSV File (N and P Tuning):** https://github.com/atabas/Major-Research-Project/blob/master/tuning_results_3.csv