The Iby and Aladar Fleischman
Faculty of Engineering
Tel Aviv University

הפקולטה להנדסה
ע"ש איבי ואלדר פליישמן
אוניברסיטת תל אביב

Using Deep Learning Methods For Identifying Breast Cancer From Mammograms

Final Report

Project Number: 17-1-1-1296

Student: Liron Kreiss 305773756

Supervisor: Dr. Dror Lederman

Location: Tel Aviv University

Introduction

Breast cancer is one of the major cancers and one of the leading causes of death among women in the world [1]. The most widely used breast screening modality is mammography. Its interpretation is performed by radiologists by visual inspection, which is time-consuming and subjected to errors due to high breast density and high variability in the appearance of breast malignancy. All of these lead eventually to a relatively high rate of unnecessary biopsy tests and missed-diagnosis of malignant tumors [2,3]. Over the years, computer-aided diagnosis (CAD) systems have been developed to automatically detect breast abnormalities and assist radiologists by indicating suspicious or high-risk regions of the images [4,5].

Data-driven machine learning (ML) techniques have become of great interest in improving CAD capabilities. A key enabling advance has been the advent of "deep learning" (DL), which allows computers to build predictive algorithms based on the features that are found to best explain observed data in a generative fashion.

In this project, we will implement a CNN in order to classify mammographs as healthy or sick (Positive and Negative respectively). In Figure 1 we can see an example of a CNN ability to identify breast cancer density vs manual assessment.

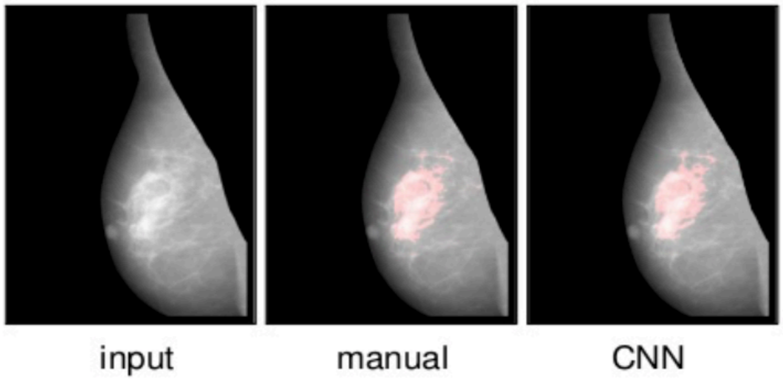


Figure1: Example of identifying breast density (one of breast cancer's causes) using CNN (Convolutional Neural Network) [6]

Project goals

This project is part of continuing efforts to develop CAD for early detection of breast cancer. Specifically, this project is aimed at utilizing deep learning methods to improve CAD performance.

Another goal in this project was to gain theoretical and practical knowledge in Deep Learning, focusing on Neural networks, mainly CNN, architecture and to acquire "Hands-on" experience in writing and implementing CNN for image classification, focusing specifically on learning Tensor Flow, one of the most popular frameworks for DL implementation.

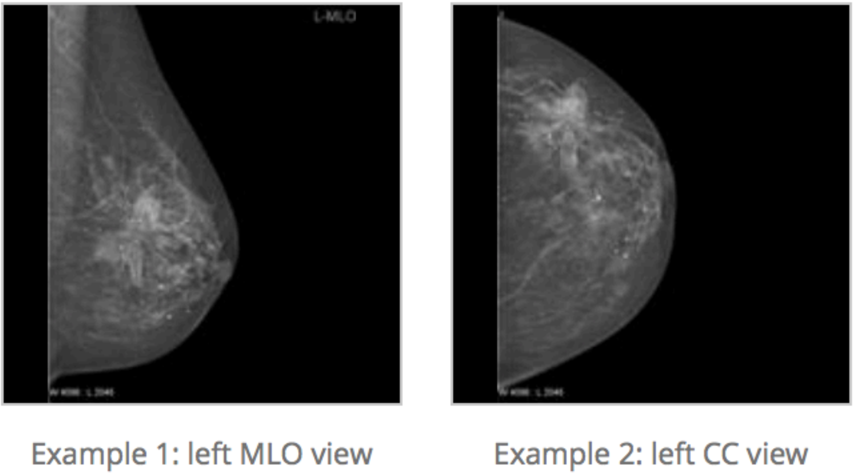
Theoretical background

Medical Background

Breast cancer is the most common cancer among women. Worldwide research efforts have been devoted to find any sort of early reliable detection method [7]. Screening mammography is the main imaging test used to detect occult breast cancer [8]. Mammography is specialized medical imaging that uses a low-dose x-ray system to see inside the breasts. A mammography exam, called a mammogram, aids in the early detection and diagnosis of breast diseases in women.

An x-ray (radiograph) is a noninvasive medical test that helps physicians diagnose and treat medical conditions. Imaging with x-rays involves exposing a part of the body to a small dose of ionizing radiation to produce pictures of the inside of the body. X-rays are the oldest and most frequently used form of medical imaging.

There are numerous mammography views, one of them is The craniocaudal view (CC view), along with the MLO view, is one of the two standard projections in a screening mammography. It must show the medial part as well the external lateral portion of the breast as much as possible [9]. The mediolateral oblique view (MLO) is taken from an oblique or angled view. During routine screening mammography, the MLO view is preferred over a lateral 90-degree projection because more of the breast tissue can be imaged in the upper outer quadrant of the breast and the axilla (armpit) [10].



Example 1: left MLO view

Example 2: left CC view

figure 2: examples of MLO and CC views [9]

Millions of mammographs are performed every year with the goal of enabling improvement treatment outcomes and longer survival times for breast cancer patients via early detection. Mammogram interpretation requires the expertise of a highly trained radiologist, which needs to be qualified to diagnose. The interpretation also can be time-consuming and is prone to interpretation variability [11]. The interpretation process includes detecting abnormal areas of density, mass, or calcification that may indicate the presence of cancer. Computer aided detection (CAD) is a technology designed to decrease observational oversights—and thus the false negative rates—of physicians interpreting medical images. The CAD system highlights the abnormal areas on the images, alerting the radiologist to carefully assess this area.

Recently, the deep learning techniques have been introduced to the medical image analysis domain with promising results on various applications such as object recognition in natural images. This success has prompted a surge of interest in applying deep convolutional networks (CNN) to medical imaging [8]. Using Deep learning techniques and neuron networks in analyzing medical images such as mammograms, require understand the feature extraction and classification process that the network produces (as an alternative to a radiologist assessment).

Deep learning and CNN

Since 2012, deep convolutional neural networks (CNN) have been a tremendous success in image recognition, reaching human performance. These methods have greatly surpassed the traditional approaches, which are similar to currently used CAD solutions. Deep CNN have the potential to revolutionize medical image analysis [12].

Image classification is the task of taking an input image and outputting a class (a cat, dog, etc.) or a probability of classes that best describes the image. For humans, this task of recognition is one of the first skills we learn from the moment we are born and is one that comes naturally and effortlessly as adults. These skills of being able to quickly recognize patterns, generalize from prior knowledge, and adapt to different image environments are ones that we do not share with our fellow machines.

A more detailed overview of what CNNs do would be that you take the image, pass it through a series of convolutional, nonlinear, pooling (downsampling), and fully connected layers, and get an output. The output can be a single class or a probability of classes that best describes the image.

These features can be straight edges, simple colors, and curves. The features are calculated based on convolution operations, we iterate these filters over the entire image. There are other layers that are interspersed between the conv layers which provide nonlinearities and preservation of dimension that help to improve the robustness of the network and control overfitting [13].

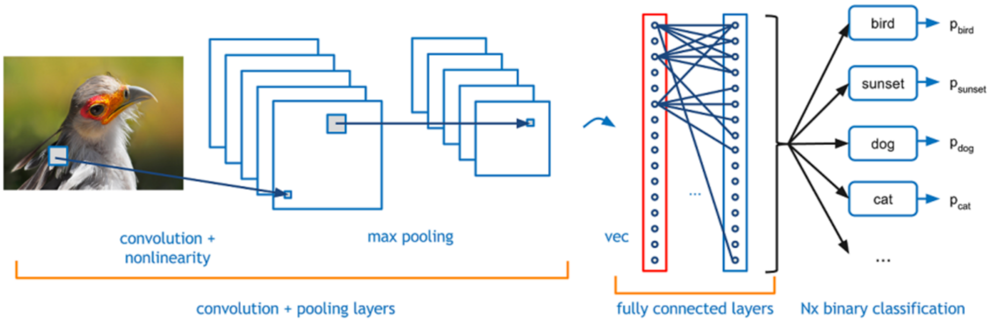


Figure 3 – simplified structure of convolutional neural network [13].

General Steps in building a network

1. Data gathering and arrangement – Accurate, actionable, accessible data is the lifeblood of any successful model. In classification problems we

usually want our data balanced (meaning similar amount of data for each class). We also need to split our data to Training set, validation set and test set for Cross-Validation. The motivation for using cross-validation is mainly to avoid overfitting of our model and help us evaluate more objectively the results.

2. Data pre-processing – Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing. In computer vision and image classification example for pre-processing of the data would be to crop, add/reduce noise from images etc.
3. Building the model – Build the network's architecture, initial and set values for hyper-parameters such as learning rate, loss functions, batch sizes, etc.
4. Model evaluation- Model Evaluation is an integral part of the model development process. It helps finding the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. One of the mode popular method to evaluate our model is the cross validation method which we mentioned earlier.

   In general, there are several metrics we are taking into consideration when evaluating a model:

- FPR – False Positive Rate, meaning the rate of "good/healthy" predictions which in fact "bad/ill" out of all results.
- TPR – true positive rate, meaning the rate of "good/healthy" predictions which indeed own this classification out of all results.
- ROC - receiver operating characteristic curve, created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.
- AUC - Accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test; an area of .5 represents a worthless test
- Precision – The rate of TPR out of all our "good/healthy/positive" predictions.

- Recall – The rate of TPR out of all actual "good/healthy/positive" results.

5. Test and present final results - When our model is trained and we feel confident with it, we can run our test set and test the model performance on data "It had never seen".

Transfer learning and Inception V3

When we consider classifying images, we often opt to build our model from scratch for the best fit, but building a deep learning model can take weeks, depending on our training dataset and on the configuration of our network, not to mention the cost of resources. Transfer learning is the ability to take advantage of the existence of a model for a custom image classification. This method prove itself mainly because of the architecture of Neural networks. In a neural network, neurons are organized in layers, different layers may perform different kinds of transformations on their/ inputs. Signals travel from the first layer (input), to the last one (output), possibly after traversing the layers multiple times as a consequence of the back propagation method. As the last hidden layer, the "bottleneck" has enough summarized information to provide the next layer which does the actual classification task.

In our work we used the pre-trained Inception V3 network, which is the model that Google Brain Team has built for image classifications and showed outstanding results. The main idea behind the inception V3 method is that instead of choosing one type of convolution (3X3 or 5X5 for example), we want to make at each layer, we can do all the options in parallel and concatenating the resulting feature maps before going to the next layer and the model decides which approach is best. Additionally, this architecture allows the model to recover both local feature via smaller convolutions and high abstracted features with larger convolutions [14].

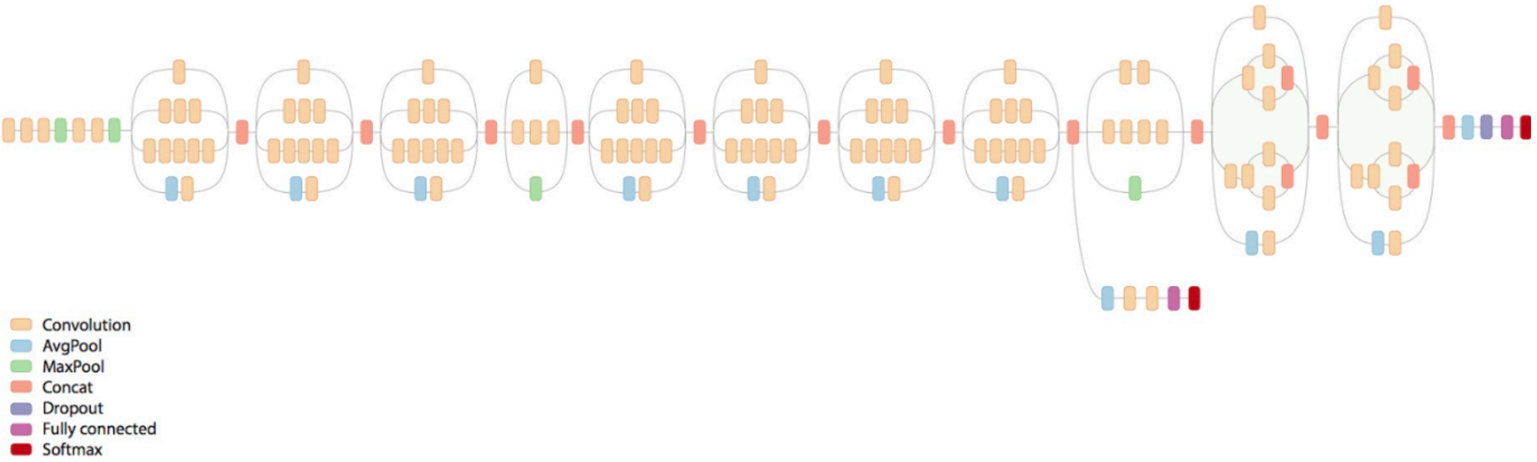


Figure 4 – Inception V3 architecture [14].

Current work

There is a lot of work in the field of CAD for identifying breast cancer from mammogrpahs, specifically using deep learning and CNNs. There are several articles about the effect of the resolution of mammogrpahs used in the training process, some present good results when keeping the image original size through all the training process [15] and others shows that cropping the mammogrpam [16] in the pre-processing stage showed good results as well. Some papers suggest use only MLO views [7], where the majority uses both MLO and CC views. There is also use of transfer learning, for example, AlexNet, one more Google's network for image classification network [16]. There is also a discussion about natural vs. control distribution of the data in terms of amount of healthy vs sick mammographs [15]. Almost every work involves visualization in evaluating and understanding the network performance [15,16].

Visualization

Although CNN can achieve great performance, it acts like a black box, and we sometimes do not understand how these networks make decisions and what information is stored by them. One way to shed light on the CNN is to visualize the network, by generating images or visual results that can in some extent reveal what information is stored by CNN.

These results could be guidelines for designing networks with higher accuracy, and could provide insight into whether they may be effective when tested on new datasets [5]. There are several methods used for this visualization, some of which are presented below. These methods are detailed in two main articles [17],[18].

Deconvnet

Since understanding the operation of a CNN requires interpreting the feature activity

in intermediate layers. Zeiler and Fergus et al [17] presented a novel way to map these activities back to the input pixel space, showing what input pattern originally caused a given activation in the feature maps. They perform this mapping with a Deconvolutional Network (deconvnet) Zeiler et al. [19]. A deconvnet can be thought of as a convnet modelthat uses the same components (filtering, pooling) but in reverse, so instead of mapping pixels to features does the opposite.

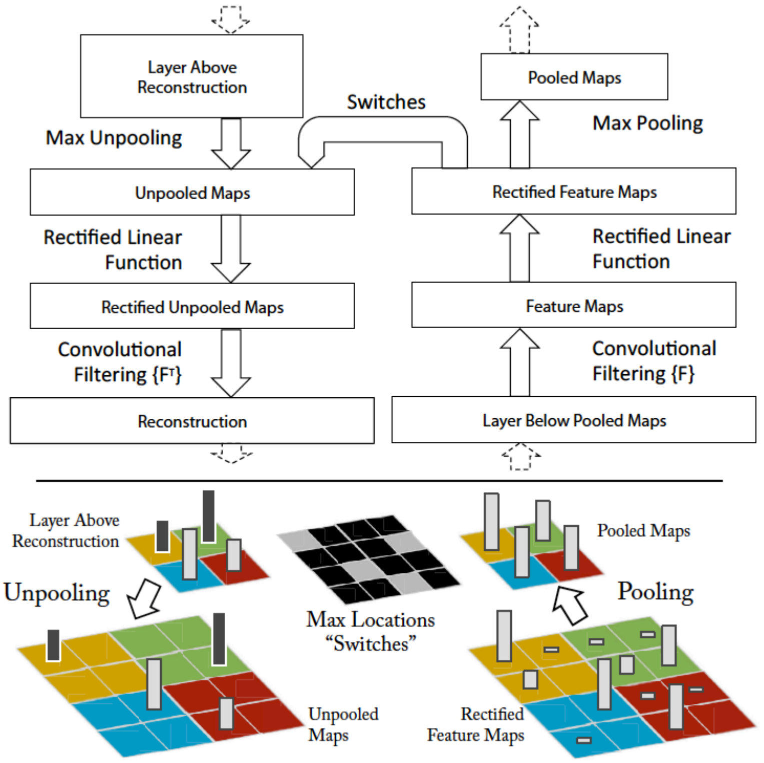


Figure 5 : taken from Zeiler and Fergus et al [17] paper - A deconvnet layer (left) attached to a convnet layer (right). The deconvnet will reconstruct an approximate version of the convnet features from the layer beneath.

Bottom: An illustration of the unpooling operation in the deconvnet, using switches which record the location of the local max in each pooling region (colored zones) during pooling in the convnet. The black/white bars are negative/positive activations within the feature map.

Image masking

Convolutional neural network visualization by image masking is a way to observe directly which parts of a image have significant influence on the final activation score. This approach is mainly applied for the situation of interpreting the decision making in scene classification tasks, where many objects with semantic meaning are contained in an image.

The image masking approach to CNN visualization has been introduced by Zhou, et al. [20]. In their work, image masking is performed by segmenting input images and masking out irrelevant region with activations below certain threshold based on feature map of a specific unit in a specific layer. The neurons with maximum activation in a feature map are first determined and then a mask is applied to this image to mask out region that has activation less than a certain threshold relative to the maximum activation. Usually, the threshold value is the maximum activation times a constant, c < 1, which may be different for each model, in order to accentuate the region with high activation. This approach is an image-centric visualization, since each visualization is generated based on a specific image and the visualization results are showed on that image.

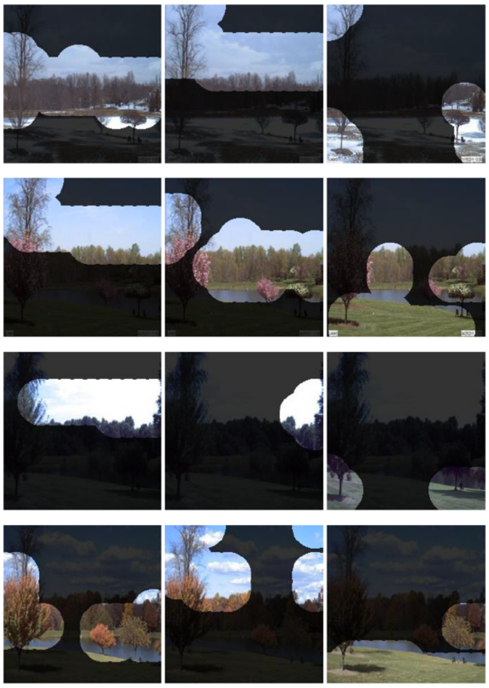


Figure 6: Example of image masking implementation [18] – classifying seasons with images from different months. January, April, July and October are selected as a representative month for each season. The row represents month and the column represents unit

<u>Tylor decomposition [18]</u>

For both classification model and regression model, the final decision making is based on the activation function. Since the final activation function is a result of summing all the activation score from previous neurons, there is an idea to directly decompose this final activation function as Taylor Series and then back-propagate these series from the last layer down to the first input layer. A heat map can be generated to show per pixel relevance, which indicates the pixel's relevance with the final activation score, as shown in figure 6 for example. If the relevance is

high for some pixels, it means that these pixels account for high contribution for the final classification decision.

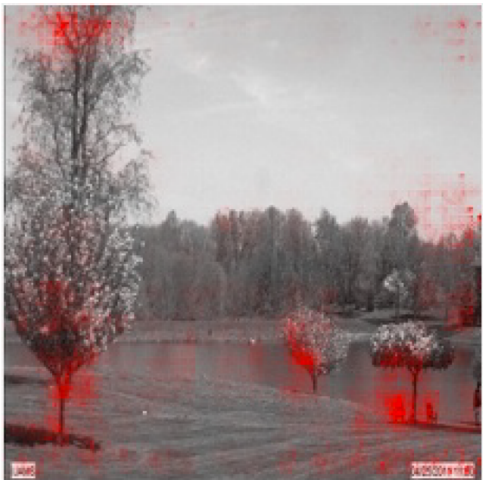


Figure 7: An example of relevance map generated by Taylor decomposition [18]

Inception

The above visualization approaches aim at showing which parts of image have important influence on the final classification result. In Inception approach, which is also called activation maximization in the work of Mahendran and Vedaldi [21], a reconstruction of the input image from image representation will be showed as the visualization of CNN. The information used to reconstruct the input image is gathered entirely from the CNN. So this method is a visualization from the perspective of visualizing what information have been stored in CNN in order to achieve some extent of accuracy.

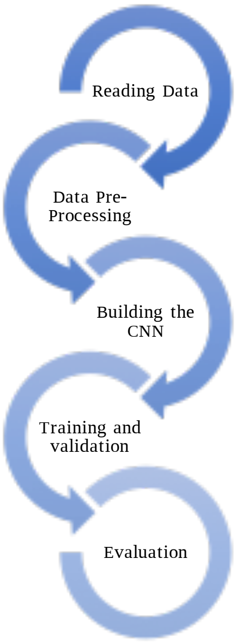
Preliminary work – preparing the infrastructure

In this project we continue to develop a scheme that was proposed in a previous study [22]. The network was written in python and at first stage we used it as is. We decided to run the network on a remote Centos (Linux) server with GPU and not locally, since the local resources we had were not enough for running the network and dealing with processing large amount of images.

We also used Tensorboard (visualization utility of Tensor Flow framework) in order to visualize the structure, weights and constants of our network. In order to view the outputs we had to expose port 6500 from the server to our localhost.

Implementation

As mentioned above, we used an existing scheme in order to classify mammograms to ″healthy″ and ″sick. The scheme was based on a CNN network which will be discussed in details. The general scheme of the work:



Reading Data

We used InBreast database which contains 385 mammograms, with MLO and CC views, both as our train and our test set, where each has a Bi-RAD value. BI-RADS is a scheme for putting the findings from mammogram screening (for breast cancer diagnosis) into a small number of well-defined categories [23].:

- 0- incomplete
- 1-negative
- 2-benign findings
- 3-probably benign
- 4-suspicious abnormality
- 5-highly suspicious of malignancy
- 6-known biopsy with proven malignancy

We have classified mammographs with Bi-RADS values of 4-6 as ″sick″ (TRUE) where all the other categories were classified as ″healthy″ (FALSE).

The data was divided into training set and test set.

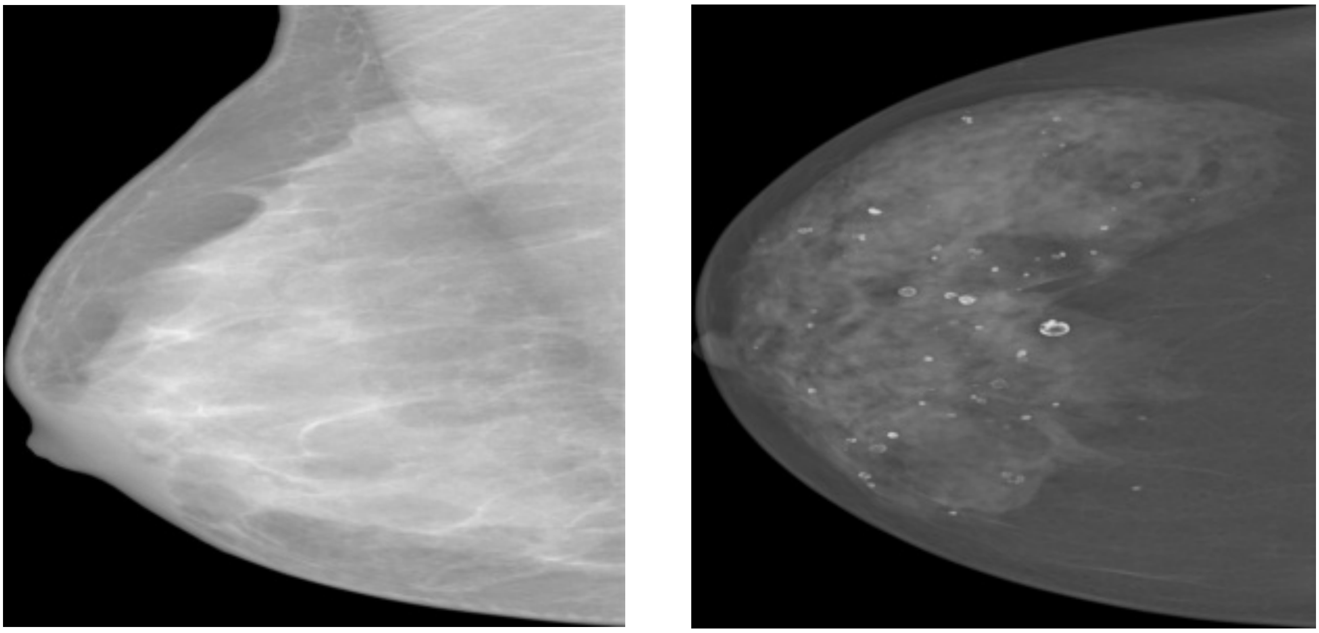


Figure 8 Left- mammogram of an healthy subject, Right – mammogram with breast cancer. Both taken from INBreasts.

Pre-Processing

As stated above, our dataset is very small. In Deep Learning models small data set can result an overfitting and/or poor results in production. In addition, Since we are dealing with binary classifications we wanted to have a relatively balanced data set, and together with the fact mentioned above about our small data set, we decided to use data augmentation method in order to enrich our dataset.

Data augmentation adds value to base data by adding information derived from internal and external sources. Our augmentation techniques focused on the image processing region and included:

- Flipping
- Add contrast
- Add Gaussian noise
- Add Brightness

The InBreast database includes 313 healthy mammograms and 75 mammograms with masses/positive findings. After augmentation, the dataset used for the model contained 3,104 "Negative/healthy" and 1,992 "Positive/sick" (the ratio is still not balanced enough, see next steps for elaboration).

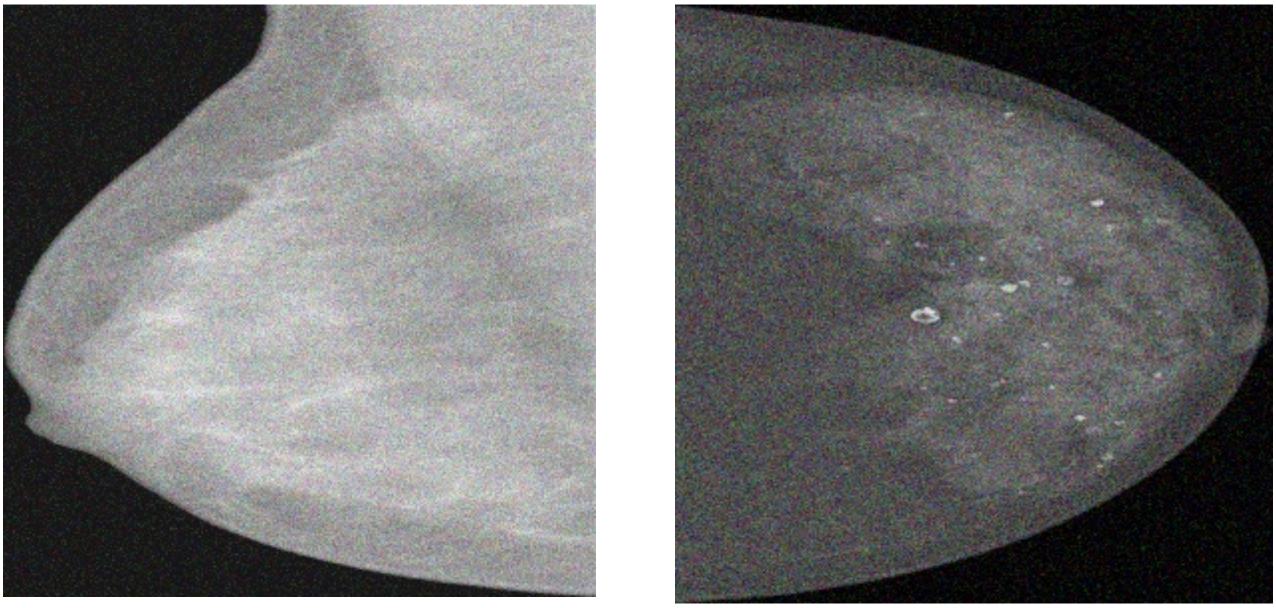


Figure 9 Left- mammogram in figure [] after adding Gaussian noise and random contrast, Right – mammogram in figure after flipping horizontally and adding Gaussian noise.

Building the network

Overall architecture of the network exported from TransorBoard:

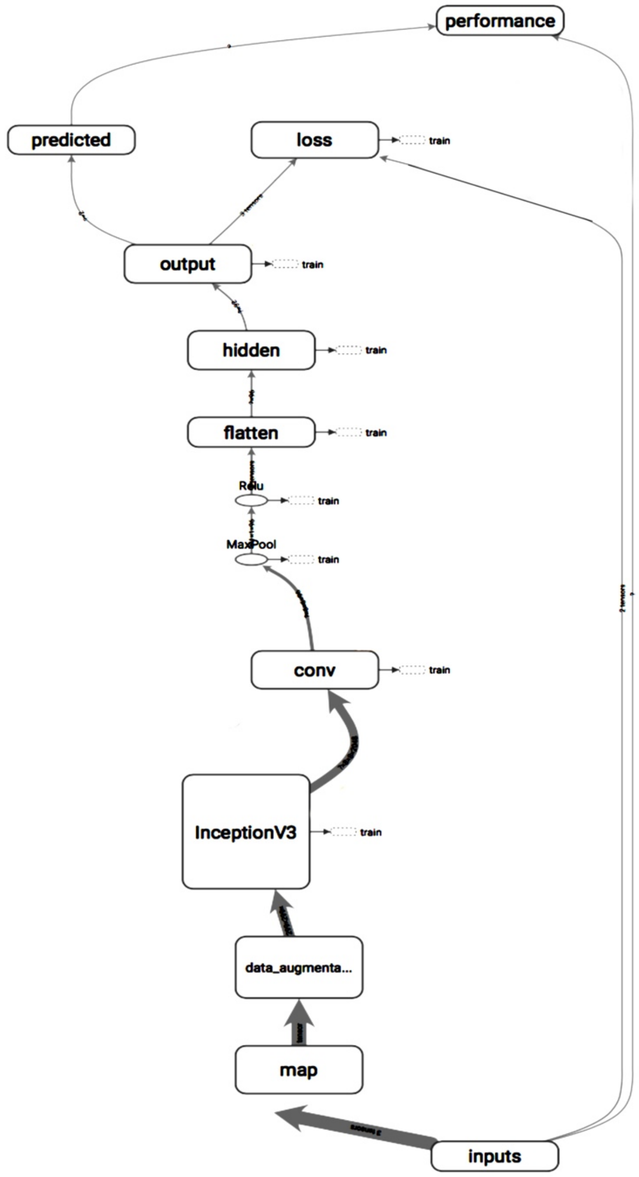


Figure 10 – Overall architecture of the network exported from TransorBoard.

Arrows represent data direction and each contains the tensor size being forwarded to the next layer.

As mentioned, we used the transfer learning technique and use the inception V3 checkpoint. Above this layer, we added a convolution layer ("conv"), added Maxpool, Relu and flatten our vector so we can use a FN (Fully connected network) at the final step. We used the cross entropy as our loss function:

$$(y \log(p) + (1 - y) \log(1 - p))$$

We chose Adam optimizer for out back projection, which is an extension to the stochastic gradient descent with learning rate[1] of 0.001. A batch size of 32 images used for the training part.

Training part

As mentioned above, we used Tensorboard in order to visualize the weights and constants of the network during the training process.

We can see that our loss during the train procedure has some peak but eventually converged, as expected.

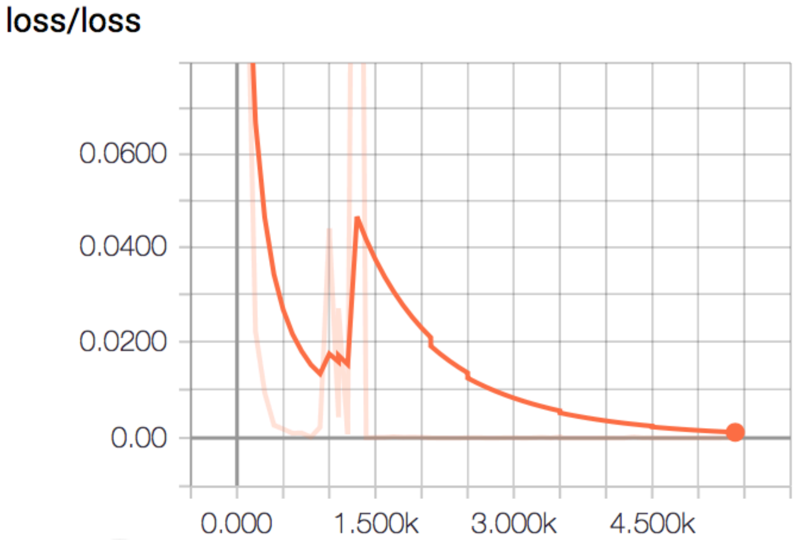


Figure 11 Loss values in each step in training process

In figure[] the weights constructed from the CNN are shown for each batch. We can immediately observer that our weights have changed in each step in the conv layer, meaning that the network did learn in each step.

[1] Learning rate is a hyper-parameter that controls how much we are adjusting the weights of our network with respect the loss gradient.

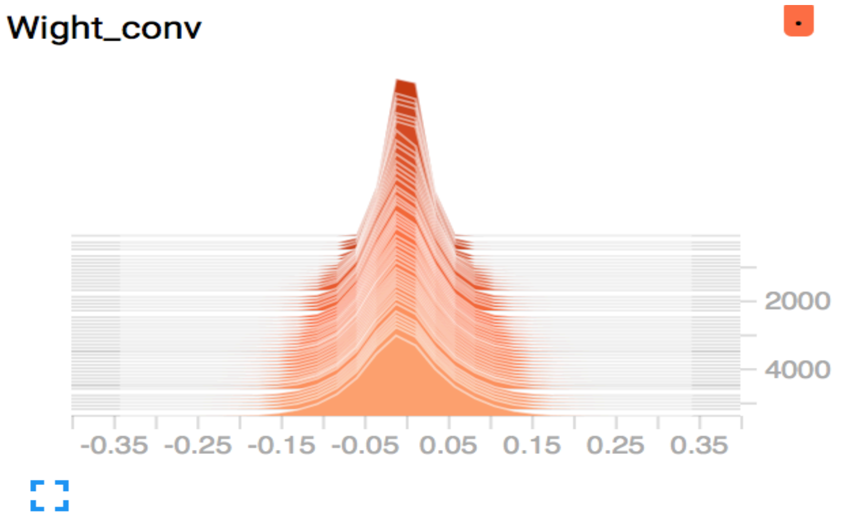
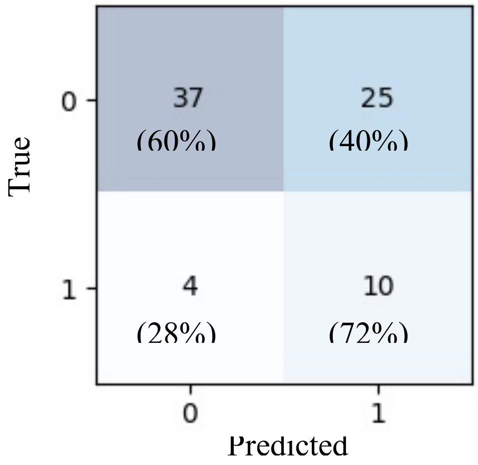


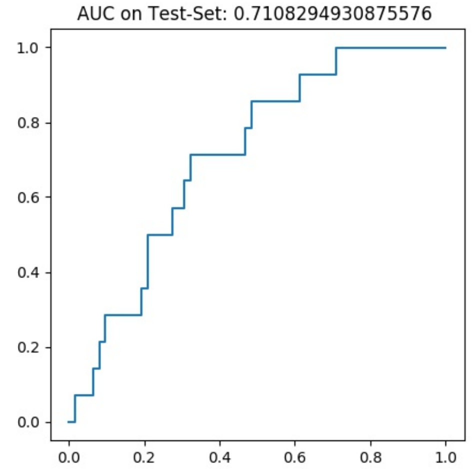
Figure 12 weights histogram in each step in training process

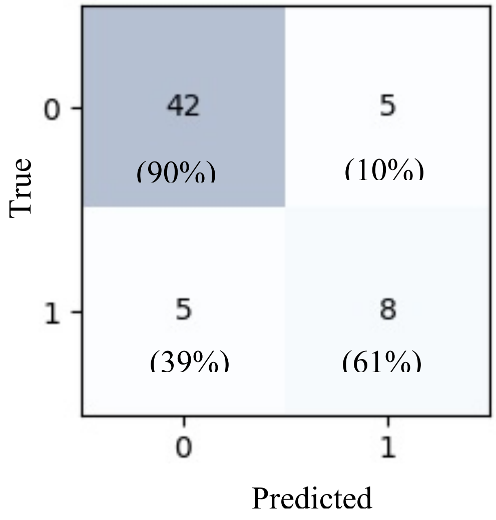
Results

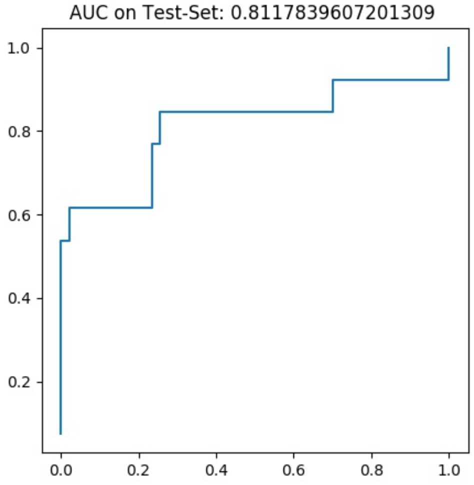
We compared the results when training the network in cross-validation (without validation set), and test it with pre-defined test set to the results training ALL data allocated for training (in all folds) and test it with test set (in all folds).

Roc, AUC and confusion matrix per fold (folds goes from 0-4 top to bottom)









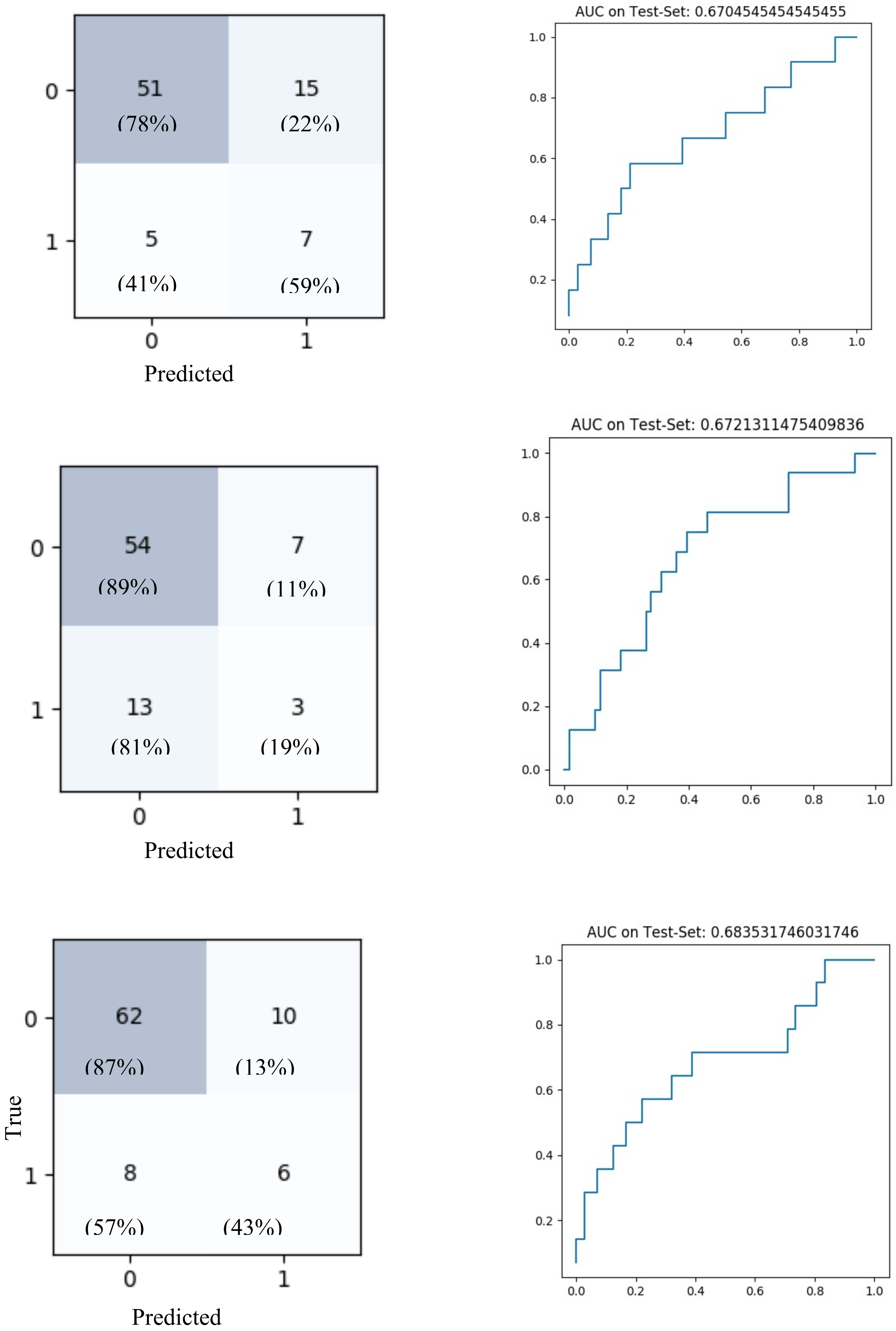


Figure 13 - ROC curves (y values are TPR, x values are FPR) with AUC values and confusion matrix for each fold

CV Average Results

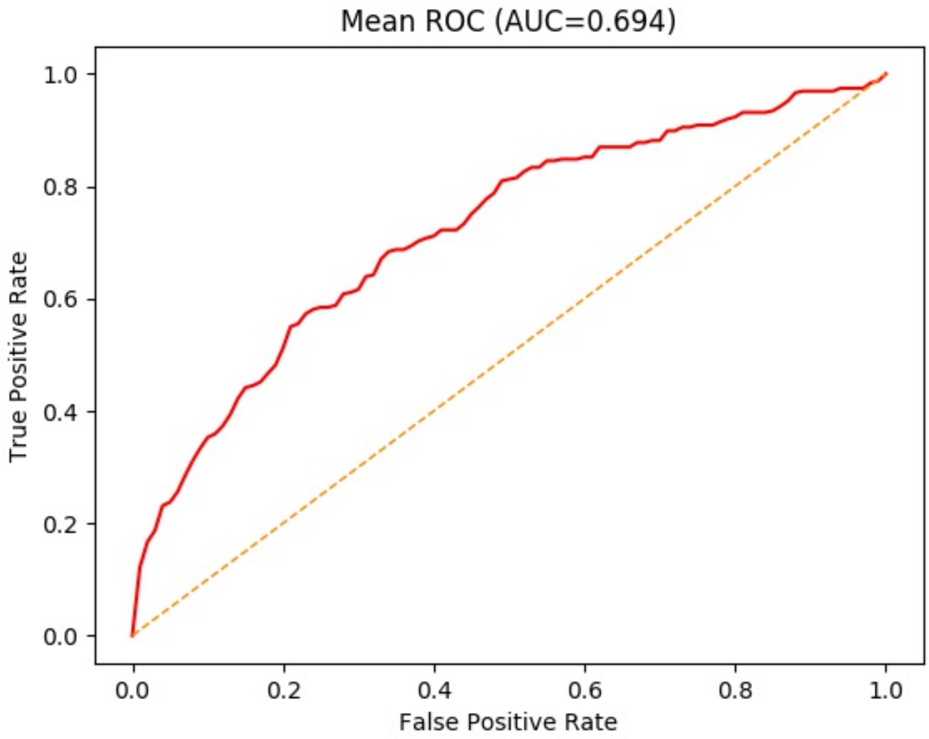
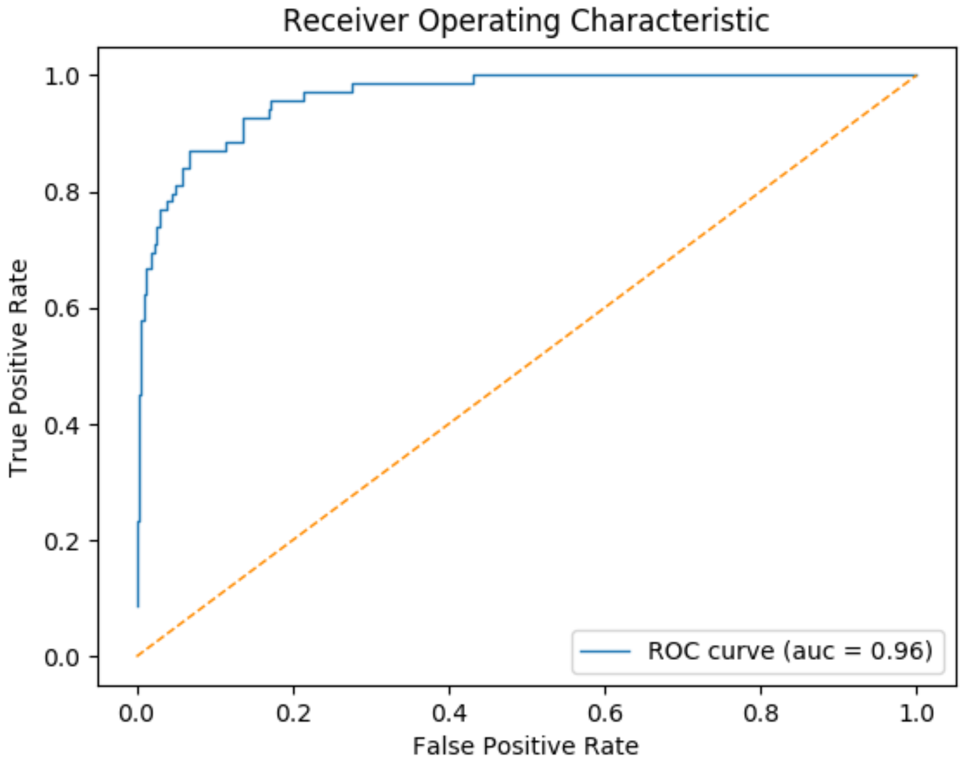


Figure 14 – Average ROC curve (y values are TPR, x values are FPR) and average AUC value

All Data Train results



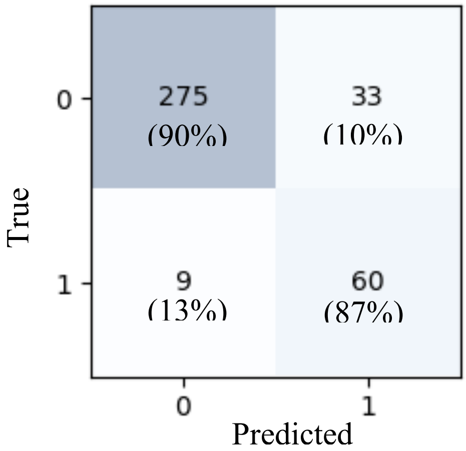


Figure 15 - ROC curves with AUC values and confusion matrix for training of all the data set

Result's assessment and discussion

In the CV section we can see that our network is having hard time classifying our "Positive" inputs while doing good job in classifying the "Negative" inputs. We can recall that our data is not balanced This can explain the network's performance.

We can also see that there is a difference between each folds' results, this usually means that our model is unstable and very hard to rely on, since there are different results for different set of data (which in "real world" we want to be able to evaluate a variety of sets).

There is a major difference in terms of AUC value between the CV and all-data train, where the results are relatively good and reliable and the network did succeed in classifying the "Negative" mammograms.

One assumption for the reason of the aforementioned behaviors is the fact that the data augmentation created a large variance between the trained data and the data being tested (which was not augmented in order to test our model with "real world" mammograms), and wasn't balanced between the folds, meaning we first split the data and then performed augmentation. Together with the fact that we have a small test set, we can conclude that in the CV part, the model was trained on data with large variance compared to the data it was tested on. When training the model on all the data however, the model is trained on a variety types of inputs, the generalization is good enough, hence the results on the test set are better.

Summary and Next steps

The project's main goal was to improve results of an existing scheme for mammograms classification. The project involved understanding the medical background, current work, specifically in CAD techniques, "zoom in" to CNN techniques, then implementing current scheme and investigate its results. As discussed in the results chapter, we saw inconsistency between the network results using CV and the results when training all the data at once. We will want to focus on that issue and understand possible root causes for this matter. We also saw that out dataset is not balanced and we have more "Negative" results rather than "Positive" results.

With all of the above, we recommend the following next steps in order to understand and improve our network results

1. Add validation set to our data and perform CV. This is the CV best practice and we want to make sure that in each training fold we are validating ourselves on different data set than the actual test set. This will help us in evaluation our model and understand it's behavior through the training process.

2. Enlarge our dataset of "Positive" mammograms using augmentation techniques in order to deal with the unbalanced data. We can try more approaches of augmentation (such as cropping) and use existing techniques currently implemented in our code when targeting only the "Positive" set. This will help out network learn more about "Positive" inputs and eventually classify them more accurately. In addition, try and change the flow of the pre-processing of the data – first implement augmentation on all train set and then split it into CV.

3. Hyperparameter tuning - Parameters which define the model architecture are referred to as **hyperparameters**, tuning hyperparametes in order to get the best results for our model is not trivial and there is a lot of research in that field. Since we are using transfer learning with pre-trained model, we can tune the CNN and FN layers we have added above it as well as the hyperparameters of our optimizer and our loss functions. CNN has a variety of modifiable hyper-

parameters such as – kernel size, padding options, strides, etc. We can also modify the learning rate, step sizes and batch size in our training process

4. Adding more visualization to the network both in the pixel domain, using the techniques mentioned in the theoretical background and in the "network" domain using more features from Tensorboard. Visulisation of the network will help us track and understand the functionality of our model – weather we are converging, overfitting, focusing on relevant features, etc.
5. We started with a very complex network since we were based on previous work. A possible approach to try and improve the results would be to start building the network architecture's from scratch and simplify the model,. For example, use 2-3 hidden layers with common architecture recommended for image classification, evaluate the results and if needed continue to more complex methods.

In addition, while working on the project we have established strong knowledge base in Deep Learning and Neural Networks. Moreover, we have acquired "hands-on" experience in building, running and evaluating a CNN for classifying images, using TensorFlow one of the most popular frameworks in implementing NN. We have also invested in exploration and understanding CNN visualization techniques.

References

[1] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2017," CA: A Cancer Journal for Clinicians, vol. 67(1), pp. 7-30, 2017

[2] X. Wang, L. Li, W. Xu, et al., "Improving performance of computer-aided detection of masses by incorporating bilateral mammographic density asymmetry: an assessment," Academic Radiology, vol. 19(3), pp. 303-310, 2011 .

[3] Zhicheng, X. Gao, Y. Wang, et al., "A deep feature based framework for breast masses classification," Neurocomputing, vol. 197, pp. 221-231, 2016.

[4] D. Lévy, J. Arzav, "Breast mass classification from mammograms using deep convolutional neural networks," arXiv:1612.00542, 2016 .

[5] E. Azavedo, S. Zackrisson, I. Mejàre, et al., "Is single reading with computer-aided detection (CAD) as good as double reading in mammography screening? A systematic review," BMC Med Imaging, vol. 12(22), Jul. 2012.

[6] Petersen, Nielsen, Diao, Karssemeijer, Lilllholm. Breast Tissue Segmentation and Mammographic Risk Scoring Using Deep Learning. Digital Mammography / IWDM, 2014

[7] Anastasia Dubrovina, Pavel Kisilev, Boris Ginsburg, Sharbell Hashoul and Ron Kimmel, "Computational Mammography using Deep Neural Networks"

[8] Krzysztof J. Geras , Stacy Wolfson, Yiqiu Shen, S. Gene Kim, Linda Moy, Kyunghyun Cho, "High-Resolution Breast Cancer Screening with Multi-View Deep Convolutional Neural Network"

[9] https://www.radiologyinfo.org/en/info.cfm?pg=mammo

[10] http://www.imaginis.com/mammography/how-mammography-is-performed-imaging-and-positioning-2

[11] Darvin Yi, Rebecca Lynn Sawyer, David Cohn the 3rd, Jared Dunnmon, Carson Lam, Xuerong Xiao, Daniel Rubin, " Optimizing and Visualising Deep Learning for Benign/Maligant Classification in Breast Tumors"

[12] Ribli,Horvath,Unger,Pollner,Casbai "Detecting and classifying lesions in mammograms with Deep Learning"

[13] https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

[14] https://hacktilldawn.com/2016/09/25/inception-modules-explained-and-implemented/

[15] Krzysztof J. Geras , Stacy Wolfson, Yiqiu Shen, S. Gene Kim, Linda Moy, Kyunghyun Cho, "High-Resolution Breast Cancer Screening with Multi-View Deep Convolutional Neural Network"

[16] Darvin Yi, Rebecca Lynn Sawyer, David Cohn the 3rd, Jared Dunnmon, Carson Lam, Xuerong Xiao, Daniel Rubin, " Optimizing and Visualising Deep Learning for Benign/Maligant Classification in Breast Tumors"

[17] Matthew D. Zeiler and Rob Fergus. Computer Vision { ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I, chapter Visualizing and Understanding Convolutional Networks, pages 818{833. Springer International

[18] Dingwen Li, Visualization of Deep Convolutional Neural Networks, Washington University in St Louis (2016)

[19] Zeiler, M., Taylor, G., Fergus, R.: Adaptive deconvolutional networks for mid and high level feature learning. In: ICCV (2011)

[20] Bolei Zhou, Aditya Khosla, _Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. CoRR, abs/1412.6856, 2014.

[21] Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks
using natural pre-images. CoRR, abs/1512.02017, 2015.

[22] Yuval Medinkov, Sapir Nehemia, Bin Zheng, Oshra Benzaquen, Dror Lederman, "Transfer Representation Learning Using Inception-V3 for the Detection of Masses in Mammography"

[23] https://breast-cancer.ca/bi-rads/