# DeepMammo
## Breast Mass Classification using Deep Convolutional Neural Networks

Arzav Jain
Stanford University
arzavj@cs.stanford.edu

Daniel Levy
Stanford University
danilevy@cs.stanford.edu

## Abstract

*Mammography is the most widely used method to screen breast cancer. Because of its mostly manual nature, the masses variability in shape and boundary as well as the low signal-to-noise ratio, a significant number of breast masses are missed or misdiagnosed. In this paper, we present multiple Convolutional Neural Network (CNN) architectures to classify pre-segmented breast masses from mammograms as benign or malignant. We test our methodology on the publicly available dataset DDSM. The best classification performance we achieve on this dataset is an accuracy of* $0.929$*, recall of* $0.934$ *and precision of* $0.924$*, successfully beating human performance. This result was achieved by modifying and fine-tuning the GoogleNet model from the ImageNet challenge.*

## 1. Introduction

Breast cancer accounts for $22.9\%$ of diagnosed cancers and $13.7\%$ of cancer related to death worldwide. In the U.S., one in eight women is expected to develop invasive breast cancer over the course of her lifetime. Routine mammography is the standard exam for preventive care and the best way (as of today) to detect breast cancer without invasive surgery. However, mammography is still a manual process, quite prone to human error due to the variable shape and size of masses [1] and their low signal-to-noise ratio, thus resulting in unnecessary biopsies or missed masses. The efficacy of such a manual process is associated with the radiologists expertise and workload [2], where a clear trade-off can be noted between sensitivity (Se) and specificity (Sp) in manual interpretation, with a median Se of $83.8\%$ and Sp of $91.1\%$ [2].

The main goal of this paper is to evaluate Deep Convolutional Neural Networks (CNNs) in classifying breast masses as benign or malignant, not according to radiologists' diagnoses but according to the pathology proven outcome (such as via ultrasound or biopsy) of the masses. Such a system could work as a second opinion for many radiologists in clinical practice as well as reveal interesting insights about discriminative features in benign versus malignant masses.

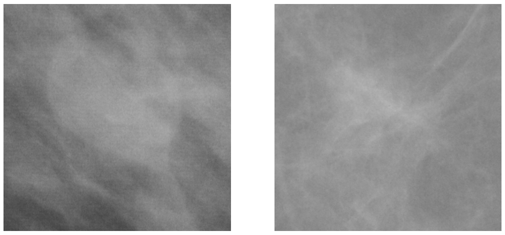
## 2. Related Work

Significant work has been done regarding mass detection using state of the art methods (namely R-CNN and random forests) as we can see in [3], [4]. Carneiro et al. consider the problem of classification on the entire mammogram using multiple views of the breast as input [6]. Classification of lesions as masses versus calcification as well as classifying masses according to the radiologist's diagnosis (encoded as BIRADS codes 0-6 on the spectrum of normal to malignant) has also been fairly treated (see [5]). However, to our knowledge, directly classifying pre-detected masses according to the final proven outcome (malignant or benign) using deep learning techniques has not been attempted.

## 3. Dataset

Our dataset comes from the Digital Database for Screening Mammography (DDSM) [7], a collaboratively maintained public dataset at the University of South Florida. It includes approximately 2500 studies each comprising both the mediolateral oblique (MLO) and craniocaudal (CC) views of each breast. Each of these images is grayscale in `.tif` format (one 16-bit channel) and is accompanied by a mask denoting the pixels that make up the pre-segmented mass if one exists.

We only consider the mammograms containing masses with their masks. This resulted in 1820 images from a total of 997 patients (see Figure 1 for examples). These images were then randomly split by patients into **training**, **testing** and **validation** (respectively $80\%, 10\%$ and $10\%$ of the total dataset). More specifically, there were 1456 images from 807 patients in the training set, 182 images from 94 patients in the validation set and 182 images from 96 patients in the test set. The validation and testing sets were each constrained to have an equal number of benign and malignant images so that an accuracy of 50% is expected by a classifier that predicts by random chance. Consequently, the training set had a class balance of 777 benign masses and 679 malignant masses.



(a) Benign (b) Malignant

Figure 1: Sample breast mass images fed as input: (a) benign with a well-defined margin and an oval shape; (b) malignant with a microlobulated margin and an irregular shape.

### 3.1. Preprocessing

In order to create this mass image dataset ready for use in our models, we first extract the mass from the full mammogram by taking a bounding box around the pixel-level mask applied to the original image. Since the context around the mass is relevant in a radiologist's diagnosis, we explored two different approaches in extracting this context:

1. **Fixed padding** of $50$ pixels all around the mass in order to achieve the same context size for all masses regardless of mass dimensions.
2. **Proportional padding** around the mass by extracting two-times the size of the mass bounding box. Bigger masses will thus have more pixels of context extracted and proportionately so as compared to smaller masses.

Since the pre-trained networks that we fine-tune take as input RGB images with 3 channels, we simply replicate our grayscale image across the 3 channels. At training time, we also perform mean subtraction with a mean image computed over the entire training set.

### 3.2. Data Augmentation

Due to the small size of our training set as well as the clear dissimilarities between mammogram images and ImageNet images, we augmented our dataset to facilitate fine-tuning of pre-trained models. Since masses do not have a particular orientation and can be expected to be seen in all configurations, performing augmentation with the transformations listed below should not alter the pathology (and hence the label) of the mass.

We applied the following transformations offline:

- **Rotating**: random rotations by angles in the interval $0 \leq \theta \leq 360$. The resulting white corners were filled with the mean-pixel value of the training set.
- **Cropping**: the images were resized to $224$ pixels along their shorter side (resizing the other dimension proportionately), after which random $224 \times 224$-sized crops were sampled from the resized image.

For each image in the training set, we performed 5 random rotations and sampled 5 random crops for each rotation, thus effectively multiplying the training set size by 25. Our overall training set size was consequently 36,400 comprising 19,425 benign and 16,975 malignant masses. For both the unaugmented and augmented datasets, we also perform random mirroring of input images online at training time.

## 4. Methods

Listed below are the architectures we experimented with as well as the training and fine-tuning strategies we tried. All methods were implemented with Caffe [12] on an NVIDIA GRID K520 GPU hosted on Amazon Web Services.

### 4.1. Shallow CNN: LevyNet

One simple architecture we experimented with as a baseline is a shallow CNN with the following layers:

- Input layer
- Convolution ($32$ $3 \times 3$ filters) - Batch Norm - ReLU - Max Pooling
- Convolution ($32$ $3 \times 3$ filters) - Batch Norm - ReLU - Max Pooling
- Convolution ($64$ $3 \times 3$ filters) - Batch Norm - ReLU - Max Pooling
- Fully-connected layer of dimension $128$ - ReLU
- Fully-connected layer of dimension $64$ - ReLU
- Fully-connected layer of dimension $2$
- Softmax

This shallow architecture was inspired by both the first few layers of AlexNet [9] and [5]. We added the batch normalization (described in [13]) to facilitate training as it was trained from scratch. We also used Xavier initialization described in [14]. This network was trained using a learning rate of $10^{-3}$ with Adam [15] and a "step" learning policy with $\gamma = 0.1$. We used a batch size of $64$. This model is henceforth refered to as **LevyNet**.

### 4.2. AlexNet

The original AlexNet from [9] was fine-tuned on both the original and augmented datasets. The architecture remains unchanged from [9], except for the last fully-connected layer which was replaced to output 2 classes instead of the 1000 ImageNet classes. We chose a batch size of 128 for all AlexNet models since that was the largest batch for which the memory requirements were met by our GPU. The learning rate multiplier for all layers was set to $0.1$ times the original value except for the last fully-connected layers which is learned from scratch from a random Gaussian initialization. We used the Adam learning rate schedule with a base learning rate of $10^{-3}$, a $L_2$-regularization penalty of $5 \times 10^{-3}$ and dropout of $0.5$ so as to not overfit the training set that is much smaller compared to the original ImageNet dataset.

We fine-tuned different AlexNet models on the three different datasets:

1. **AlexNet (No Aug-Small Context)**: the unaugmented dataset with a fixed-size padding.
2. **AlexNet (No Aug-Large Context)**: the unaugmented dataset with a proportionally sized padding.
3. **AlexNet (Aug-Large Context)**: the augmented dataset with a proportionally sized padding.

### 4.3. GoogleNet

We modified the GoogleNet architecture from [11] by changing the last fully-connected layer to output 2 scores and removing the two auxiliary classifiers. Although these classifiers were present in the original architecture to combat the vanishing gradient problem and provide regularization, we found that loss convergence was much faster without them.

We considered two parameter variations on the resulting architecture:

1. **Shallow Training**. In order to train the last inception module faster than the previous layers, the learning rate multiplier for the *inception_5b* layers was the same as that in the original network while the previous layers had a $0.1$ learning rate multiplier. Dropout was reduced from $0.4$ to $0.1$ since masses were very localized to the center of the image and hence ignoring neurons whose perceptive field included the mass would only hurt the classification score. Instead, to avoid overfitting the $L_2$-regularization penalty was increased to $5 \times 10^{-4}$.
2. **Deeper Training**. In addition to the *inception_5b* module, the learning rate multiplier for the *inception_5a* module was also kept the same as that in the original network. This allowed deeper fine-tuning of the convolution layers in these last two inception modules to better learn high-level features in mammogram images. To avoid overfitting, dropout was increased to $0.2$ as compared to shallow training and the $L_2$-regularization penalty was chosen to be $10^{-3}$.

In both variations, the base learning rate was set to $10^{-2}$, the batch size was set to 32 (the largest that could fit in memory) and the learning rate multiplier for the last fully-connected layer was set to 10 and 20 for the weights and bias respectively to facilitate aggressive learning of these parameters. In line with the original training process of GoogleNet, we picked vanilla SGD (Stochastic Gradient Descent) as the learning rate schedule with a polynomial decreasing policy at a power of $0.5$.

## 5. Experiments and Results

Described below are the experiments and results from various models described in section 4.

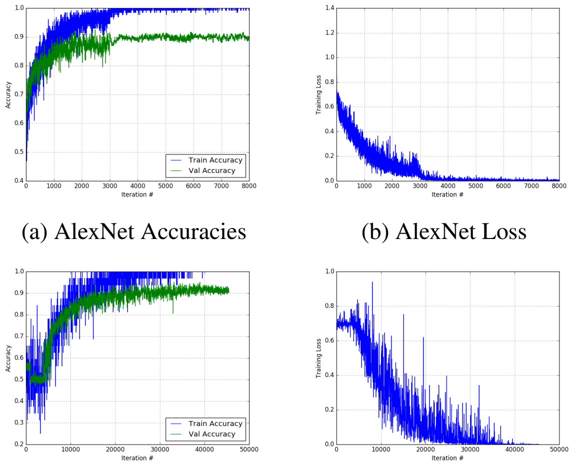
**Evaluation Metric**. As is often the case in medical applications, **recall** and (to some extent precision) are key to measuring performance. In the case of mammogram screening, we wish to greatly reduce the number of false negatives (patients with malignant masses falsely classified as benign; in clinical practice, such patients would go untreated) as compared to the number of false positives (patients with benign masses falsely classified as malignant; in clinical practice, such patients would only need an additional biopsy to confirm the benign mass and hence, the cost of misclassification is low).

### 5.1. Learning Process

In Figure 2, we present the loss and train-val accuracies for our top two models, namely AlexNet (Aug-Large Context) and GoogleNet (Deeper Training), to get more insight into the learning process. Both the models successfully converged to a very low loss albeit they both overfit the training data. The greater noise in the GoogleNet loss and accuracies is due to the vanilla SGD learning rate schedule; AlexNet was instead fine-tuned using Adam [15] giving a relatively smoother plot. Note that AlexNet converged at about 10 epochs whereas GoogleNet took about 35 epochs to converge, again probably due to the slower SGD learning rate schedule.

### 5.2. Fixed Padding vs Proportional Padding

As discussed in section 3.1, we explore the role of context around the breast mass in aiding classification. To better un-



(a) AlexNet Accuracies (b) AlexNet Loss

(c) GoogleNet Accuracies (d) GoogleNet Loss

Figure 2: Learning and fine-tuning process for our top two models: AlexNet (Aug-Large Context) and GoogleNet (Deeper Training). Shown above are train/val accuracies on the left and loss on the right.

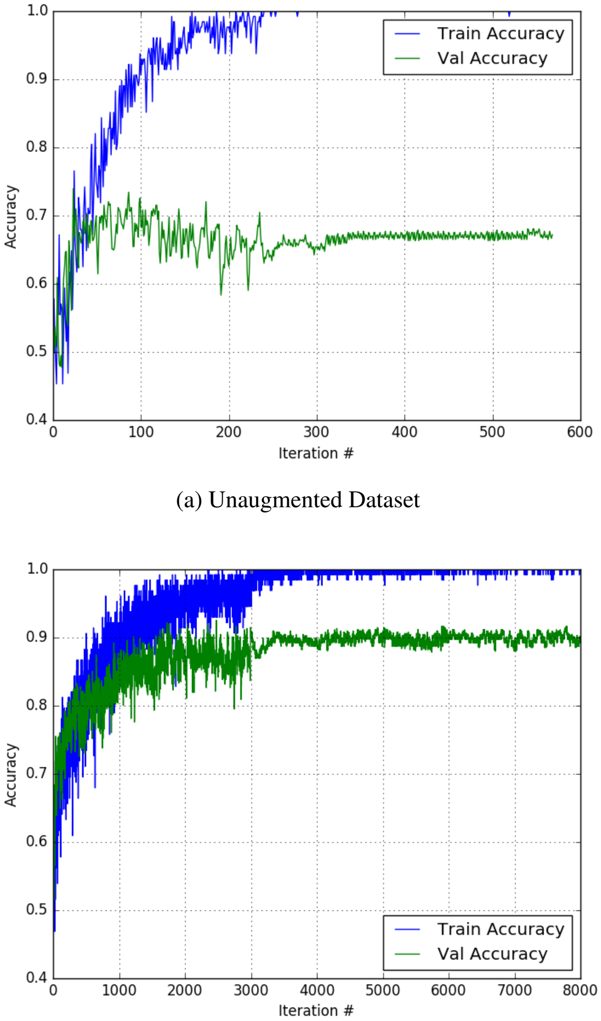
derstand whether more context around the mass helps in discriminating benign from malignant masses, we fine-tuned AlexNet on two different datasets - one with Fixed Padding and the other with Proportional Padding. The results are presented in Table 1. Given the increased validation accuracy, we find that taking a proportionately larger context around the mass does encode some information about the pathology of the mass. Consequently, we use proportional padding for the rest of our paper.

| Model | Validation Accuracy |
|---|---|
| AlexNet(No Aug-Small Context) | 0.64 |
| AlexNet(No Aug-Large Context) | **0.71** |

Table 1: Influence of context around the breast mass on the model performance.

### 5.3. Augmented Dataset vs Unaugmented Dataset

The small dataset size was a major bottleneck of our problem and as such data augmentation was an attractive solution. As can be seen in Figure 3, data augmentation greatly improves AlexNet's performance on the validation set. A deep network such as AlexNet quickly overfits the small training set in the unaugmented case within approximately 200 iterations ($\approx$ 20 epochs) . Consequently, it gives an accuracy of 0.67 and recall of 0.66 on the validation set. As expected, by increasing the training set size with augmentation AlexNet takes longer (about 3000 iterations $\approx$ 10 epochs) to overfit the training data while at the same time giving better accuracy of 0.90 and recall of 0.93 on the validation set.



(a) Unaugmented Dataset

(b) Augmented Dataset

Figure 3: Improvement in validation accuracy on AlexNet due to data augmentation.

### 5.4. GoogleNet: Deeper vs Shallow Training

As discussed in section 4.3, two GoogleNet models were fine-tuned with different parameters. The Shallow model converged within 20 epochs while the Deeper model converged withing 40 epochs. This makes sense since the Deeper model requires more iterations to better learn the high-level features at both the *inception_5a* and *inception_5b* modules.

Four snapshots of each model were also taken at regular intervals during fine-tuning. We show the performance of each of these snapshots on the validation set in Figure 4. Although the accuracy of the two models closely follow each other, the Deeper model does better on recall throughout whereas the Shallow model does better on precision. Note that the accuracy, precision and recall for the Deeper model is best at the 0.75 mark. Thus, we take this snapshot at 30 epochs ($= 0.75 \times 40$) to be our final Deeper model in the rest of this paper (and in particular, Table 2).

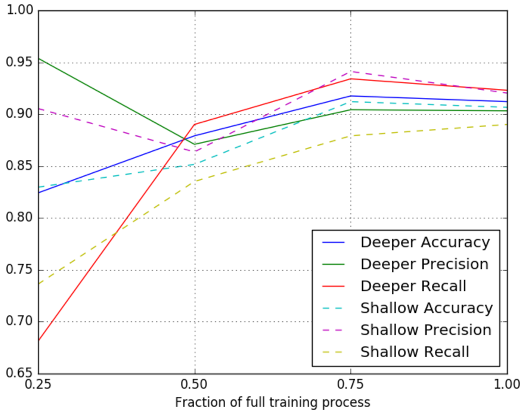


Figure 4: Validation accuracy, precision and recall for the Deeper Training and Shallow Training GoogleNet models. The x-axis represents the fraction of the total number of epochs which is 40 for the Deeper model and 20 for the Shallow model.

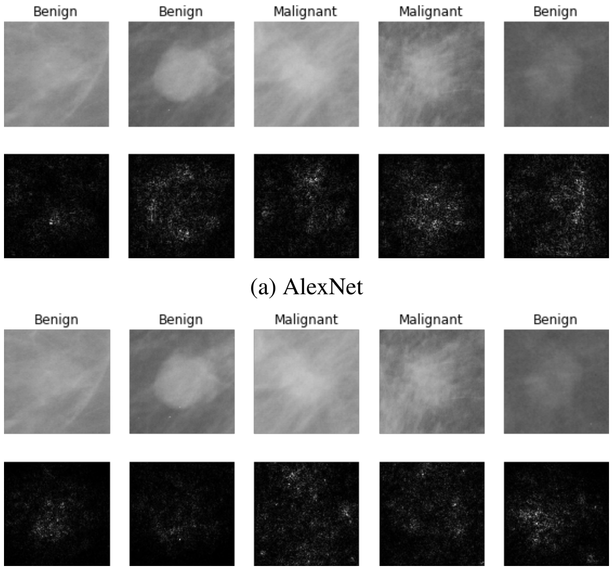
## 5.5. Visualizations

### 5.5.1 Saliency Maps

In order to get deeper insights into how the network is making classifications and which regions of the input image it is most sensitive to, we plot the Saliency Maps for five images from the dataset in Figure 5. The methodology is described in [16].

From Figure 5(a), we see that the outlines of the masses are clearly visible in the saliency maps. The image gradients closely track the mass shape and position; when the mass is diffuse, the saliency map is as well. This means that AlexNet successfully learns to attend to the masses while yet not completely ignoring the context around the mass.

For GoogleNet on the other hand, the saliency maps are overall more diffuse. The highlighted parts of the mass are sometimes also different; for example, for the benign tumor on the far right, GoogleNet attends more to the lower left part of the mass whereas AlexNet attends more to the top right.



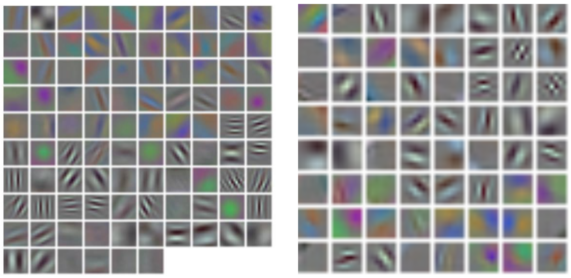
(a) AlexNet

(b) GoogleNet

Figure 5: Saliency maps for AlexNet (Aug - Large Context) and GoogleNet (Deeper Training) on five images from the validation set.

### 5.5.2 Weights visualizations

In Figure 6, we visualize the weights learned by the first layer of our networks. It serves as a sanity check; because the images are in black and white, we expect that after convergence the filters should also be in grey scale (with the caveat that this layer was trained with a tenth of the global learning rate and so the pre-trained filters from the ImageNet dataset for both networks may still show up). Figure 6 indeed shows that more than half of the filters are in grey scale which is satisfactory. The results are similar for both the AlexNet and GoogleNet. The lack of noise and the presence of nice, smooth first-layer weights for both networks indicate that both networks were trained well.



(a) AlexNet

(b) GoogleNet

Figure 6: Visualization of the first-layer filters of a trained AlexNet and GoogleNet.

| Model | Accuracy | Precision | Recall | # Epochs |
|---|---|---|---|---|
| LevyNet (Aug-Large Context) | 0.604 | 0.587 | 0.703 | 35 |
| AlexNet (Aug - Large Context) | 0.890 | 0.908 | 0.868 | 30 |
| GoogleNet (Aug - Large Context) - Shallow Training | 0.912 | 0.921 | 0.901 | 20 |
| GoogleNet (Aug - Large Context) - Deeper Training | **0.929** | **0.924** | **0.934** | 30 |

Table 2: Summary of results on the test set. All models were trained on the augmented dataset with proportional padding.

### 5.6. Conservativeness

Shown in Table 3 are the accuracies and recall of three models on the validation set. We see that recall is usually in the range of or greater than precision, thus suggesting that our models correctly classify a larger fraction of the malignant masses (fewer false negatives) than benign masses. This conservative property of our models is arguably desired given that we don't want to misdiagnose malignant masses.

| Model | Precision | Recall |
|---|---|---|
| LevyNet | 0.649 | **0.692** |
| AlexNet (Aug - Large Context) | 0.904 | **0.934** |
| GoogleNet-Shallow Training | 0.920 | 0.890 |
| GoogleNet-Deeper Training | 0.912 | **0.923** |

Table 3: Model metrics on the validation set.

### 5.7. Final results

Our final results of all models on the test set are presented in Table 2. GoogleNet with Deeper Training outperforms the other models by a fair margin (with a recall of $0.934$ compared to at most $0.901$ for the other models). GoogleNet seems more suited for fine-tuning because the inception architecture make it less prone to the vanishing gradient problem whilst keeping a very deep structure. The number of parameters of the GoogleNet is a lot smaller (5 million) compared to the AlexNet or VGGNet which have over $100$ million parameters that sometimes favor overfitting.

We also see that our best model reaches as high as $0.934$ recall which **outperforms human performance** with radiologists showing a recall between $0.745$ and $0.923$ (according to [2]). This result is very promising for real-life use of such models in clinical practice.

## 6. Conclusion

We evaluated multiple CNN models on the task of classifying breast masses as benign or malignant. Our approach validates the usefulness of data augmentation in this application as it remarkably increased performance to rival that of trained radiologists [2]. We showed that more context around the mass can be essential to classification. We also visualized a number of masses and their interpretation by our models to glean better insights into how our models make their predictions. This information can be very useful to radiologists to either confirm or deny their past intuitions as well as present novel ones. Lastly, we demonstrated how we can transfer learning from models pre-trained on the ImageNet data to a completely different domain such as mammogram images and yet achieve state-of-the-art results.

## 7. Future Work

A few approaches we would like to explore in the future include:

1. **VGGNet**. We experimented with VGGNet by taking cuts of the original architecture at each of the max-pooling layers and placing classifiers on top. However, we were unable to successfully get the loss to converge in time and hence we hope to try again in the future given the promising performance of VGGNet in other domains.

2. **Model Ensembles**. As is a common trick when working with CNNs, training multiple models and averaging their predictions at test time would greatly boost classification performance as has been found in other domains. We could use two different models such as AlexNet and GoogleNet or even different snapshots of the same model such as GoogleNet in the ensemble.

3. **Multi-view**. In this paper, we treat both the CC and MLO views of the mass indifferently and make a classification without that knowledge and separately for each view. However, as attempted in [6], taking both the CC and MLO view of the same mass together as input and subsequently performing classification would be a promising approach. Model ensembles may also be another idea here with two different CNNs, one for each view, both voting for the final score.

4. **More Data**. We obtained very satisfactory results by augmenting a small dataset of $1820$ images. However, we could greatly improve our classifier's generalizability if we were to also train on data from different

sources such as the INbreast database [8].

5. **Semantic Analysis** Clustering the last activations map (with t-SNE for example) would be a great visualization tool to see if the clusters match the semantic features used by radiologists in making a diagnosis (such as mass margin and density).

### Acknowledgements

We thank Professor Daniel Rubin in the Department of Radiology, Stanford University, for his guidance and mentorship throughout the project. We gratefully acknowledge Rebecca Sawyer's help in obtaining the dataset. We also thank Andrej Karpathy, Justin Johnson, Subhasis Das and the entire CS231N staff for helpful suggestions and feedback on our project.

## References

[1] Ball, J., Bruce, L. *Digital mammographic computer aided diagnosis (cad) using adaptive level set segmentation.* In: EMBS 2007. 29th Annual International Conference of the IEEE, IEEE (2007) 49734978.

[2] Elmore, J.G., Jackson, S.L., Abraham, L., et al. *Variability in interpretive performance at screening mammography and radiologists characteristics associated with accuracy.* Radiology 253(3) (2009) 641651.

[3] Dhungel N., Carneiro G., Bradley A. 2015. *Automated Mass Detection from Mammograms using Deep Learning and Random Forest*.

[4] Dhungel N., Carneiro G., Bradley A. 2015. *Deep Learning and Structured Prediction for the Segmentation of Mass in Mammograms*.

[5] Agarwal V., Carson C. 2015. *Using Deep Convolutional Neural Networks to Predict Semantic Features of Lesions in Mammograms.*

[6] Nascimento J., Carneiro G., Bradley A. 2015. *Unregistered Multiview Mammogram Analysis with Pre-trained Deep Learning Models*

[7] M. Heath , K. Bowyer , D. Kopans , R. Moore and P. J. Kegelmeyer. *The digital database for screening mammography*, Proc. Int. Workshop Dig. Mammography, pp.212 -218 2000

[8] Moreira, I.C., Amaral, I., Domingues, I., Cardoso, A., Cardoso, M.J., Cardoso, J.S. *Inbreast: toward a full-field digital mammographic database*. Academic Radiology, 19(2) 236248, 2012.

[9] Krizhevsky A., Sutskever I., Hinton G. *Imagenet classification with deep convolutional neural networks*, NIPS, 2012.

[10] Simonyan K., Zisserman A. *Very deep convolutional networks for Large-Scale Image Recognition*, 2014.

[11] Szegedy C. et al. *Going Deeper with Convolutions*, CVPR, 2014.

[12] Jia, Y. and Shelhamer, E. and Donahue, J. and Karayev, Sergey and Long, Jonathan and Girshick, Ross and Guadarrama, Sergio and Darrell, Trevor. *Caffe: Convolutional Architecture for Fast Feature Embedding.* arXiv preprint arXiv:1408.5093, 2014.

[13] Ioffe S., Szegedy C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, ICML, 2015.

[14] Glorot X., Bengio Y. *Understanding the difficulty of training deep feedforward neural networks*. AISTATS, 2010.

[15] Kingma D. P., Ba J. *Adam: A Method for Stochastic Optimization*. CoRR, Volume abs/1412.6980, 2014.

[16] Simonyan K., Vedaldi A., Zisserman A. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. CoRR, Volume abs/1312.6034, 2013.