# Second Look: Cancer Detection in Mammograms using Public Data Sets

Abhijit Thatte, Ankith Gunapal,
Jonathan Landesman, Andrew Lam

UC Berkeley School of Information

**Abstract**

Automated breast cancer detection faces distinct technical hurdles. Mammograms are high-dimensional (up to 4096x4096), high resolution, and are dominated by non-relevant information, mainly the components of the breast that are non-cancerous. Tumorous regions are typically small relative to the overall image - 0.5% and 1.2% of the overall image on average, for calcifications and masses, respectively, and can also be highly varied. In addition, in the real world there are many more negative cases than positive, limiting prospects for generalization. Finally, sizeable public data sets are unavailable, posing additional training challenges.

To address these issues, the Second Look team set out to explore whether Generalized Adversarial Networks (GANS) could create an augmented dataset that allows for greater accuracy on public mammogram datasets.[1] Following our inability to successfully generate new mammograms and/or mass lesions, we implemented an innovative architecture, Mask R-CNN, to localize masses. We additionally attempted to recreate state of the art mammogram detection using sliding windows across 256x256 slices of the image.

Ultimately our results compare with state of the art. Using a small training size, our methodology achieves 0.73 AUC on each individual slice of a mammogram (a 'patch') and 0.93 AUC when diagnosing an entire image.

**Related Work**

Computer aided diagnosis ("CAD") products, based on early artificial neural networks, have been in production since at least the late 1990s [1]. Jalalian et al. review pre-2012 (and pre-Alexnet) work, and report ROC scores of up to 0.98, but always on extremely limited datasets, ranging from 100-500 mammograms [2]. More recent work, such as including Levy and Jain [3], leveraged the DDSM dataset with predefined regions of interest ("ROIs") to determine whether a given, already identified region was benign or malignant. These papers did not attempt full scale processing of the entire mammogram. State of the art results on publicly available datasets (~10,000 images) are achieved by Ertosun and Rubin [4] and Lotter et al [5] using similar strategies: first the data are divided up into rolling windows ("patches")

[1] Second look is a team of four MA students at UC Berkeley's Masters of Information and Data Science (MIDS). Team members are: Ankith Gunapal, Andrew Lam, Abhijit Thatte and Jonathan Landesman

and then the researchers proceed to classify each patch into benign, malignant, or no tumor, and then follow several secondary processing steps.

Keras et al [6] and Therapixel [7] similarly use a patches approach, but also had access to much larger datasets (886,00 and 647,000 images), and achieve AUCs of 0.81 and 0.87 respectively. Of note, Lotter et al [5] reports an AUC of 0.92 on the public DDSM dataset, but only achieved an AUC of 0.84 on the larger dataset, also used by the Therapixel [7] in the DREAM challenge.

Other related work includes Yi et al. reported an accuracy of 85% when using an ensemble of 100 parallel networks using GoogLeNet architecture [8], Levy et al. reported an accuracy of 92.9% when using a GoogLeNet architecture on breast masses pre-detected by radiologists [9], and Bekker et al. reported an accuracy of 78.7% with a neural network that needs images of both craniocaudal (CC) and mediolateral-oblique (MLO) views [10].

None of the above papers attempted to segment the image to identify the exact location and spread of the tumor. Dhungel et al. reported a dice index of 88% by using Structured Support Vector Machines and Deep Belief Networks[11]. However, they did not attempt to classify the tumors.

**Data**

Mammographic Image Analysis Society (MIAS) database [12] and Digital Database for Screening Mammography (DDSM) [13] are the two dominant publicly available mammogram Databases, composed of 322 and 10,239 images respectively. These images range in size from 1024^2 to approximately 4096^2.

**General Adversarial Networks (GAN)**

The total dataset size of 10,561, high resolution images, is an order of magnitude lower than standard image recognition competition datasets (MNIST and CFAR-10, at 60,000 images each.) Our first experiment, therefore, was to explore the use of General Adversarial Networks (GAN) in order to generate more images.[14] Our primary tool for investigation was DCGAN, as it has become a standard GAN baseline.

The training set was split into 4000 patches of 256 x 256 across three classes of images . The generator and discriminator were designed as shown in Figure 1a and 1b respectively.

| | | |
|---|---|---|
| dense_1 (Dense) | (None, 8192) | 32776192 |
| leaky_re_lu_1 (LeakyReLU) | (None, 8192) | 0 |
| batch_normalization_1 (Batch | (None, 8192) | 32768 |
| reshape_1 (Reshape) | (None, 8, 8, 128) | 0 |
| up_sampling2d_1 (UpSampling2 | (None, 16, 16, 128) | 0 |
| conv2d_1 (Conv2D) | (None, 16, 16, 64) | 204864 |
| leaky_re_lu_2 (LeakyReLU) | (None, 16, 16, 64) | 0 |
| batch_normalization_2 (Batch | (None, 16, 16, 64) | 256 |
| up_sampling2d_2 (UpSampling2 | (None, 32, 32, 64) | 0 |
| conv2d_2 (Conv2D) | (None, 32, 32, 32) | 51232 |
| leaky_re_lu_3 (LeakyReLU) | (None, 32, 32, 32) | 0 |
| batch_normalization_3 (Batch | (None, 32, 32, 32) | 128 |
| up_sampling2d_3 (UpSampling2 | (None, 64, 64, 32) | 0 |
| conv2d_3 (Conv2D) | (None, 64, 64, 16) | 12816 |
| leaky_re_lu_4 (LeakyReLU) | (None, 64, 64, 16) | 0 |
| batch_normalization_4 (Batch | (None, 64, 64, 16) | 64 |
| up_sampling2d_4 (UpSampling2 | (None, 128, 128, 16) | 0 |
| conv2d_4 (Conv2D) | (None, 128, 128, 8) | 3208 |
| leaky_re_lu_5 (LeakyReLU) | (None, 128, 128, 8) | 0 |
| batch_normalization_5 (Batch | (None, 128, 128, 8) | 32 |
| up_sampling2d_5 (UpSampling2 | (None, 256, 256, 8) | 0 |
| conv2d_5 (Conv2D) | (None, 256, 256, 1) | 201 |
| activation_1 (Activation) | (None, 256, 256, 1) | 0 |

Fig 1a

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_6 (Conv2D) | (None, 128, 128, 32) | 832 |
| leaky_re_lu_6 (LeakyReLU) | (None, 128, 128, 32) | 0 |
| dropout_1 (Dropout) | (None, 128, 128, 32) | 0 |
| conv2d_7 (Conv2D) | (None, 64, 64, 64) | 51264 |
| leaky_re_lu_7 (LeakyReLU) | (None, 64, 64, 64) | 0 |
| dropout_2 (Dropout) | (None, 64, 64, 64) | 0 |
| flatten_1 (Flatten) | (None, 262144) | 0 |
| dense_2 (Dense) | (None, 1) | 262145 |

Fig 1b

Fig 2a and FIg 2b show sample images generated for 50 epochs and 1000 epochs

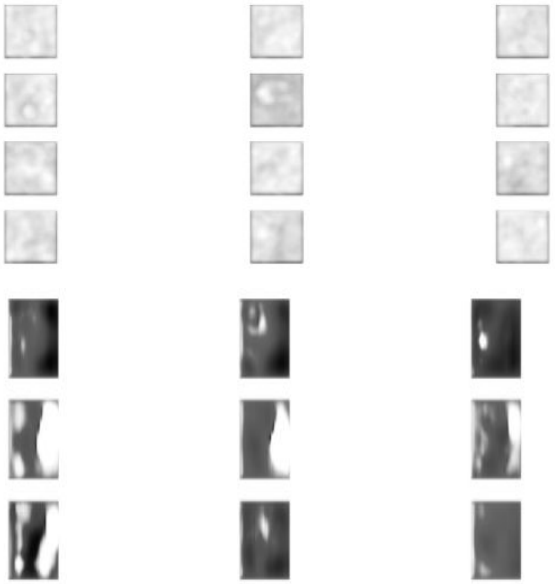

Fig 2a

Fig 2b

The clarity of these images are clearly insufficient for breast cancer diagnosis. As a validation check, training on 60k patches of GAN generated images and validating on real images resulted in an accuracy of 5% after 25 epochs.

**Mask-RCNN**

R-CNN is an approach to attempt to localize the components of an image that account for the classification, and highlight those regions. R-CNN uses selective search [15] to extract region proposals from an input image. Then R-CNN resizes the region to a square size and passes it through a modified version of AlexNet [16]. R-CNN adds a Support Vector Machine (SVM) on the final layer of CNN for classification.

Finally, R-CNN runs the bounding boxes through a linear regression model to output tighter coordinates once the object has been

classified.

R-CNN has high accuracy, however R-CNN is notoriously slow because it runs all region proposals (2000 per image) through the CNN. In addition, the pipeline of CNN to generate image features, SVM for object classification and linear regression for tightening bounding boxes is extremely hard to train.

He et al. created Mask R-CNN by extending a version of R-CNN known as "Faster R-CNN" by adding a branch for predicting class-specific object mask for Instance Segmentation in parallel with the existing object classifier and bounding box regressor.[17]

We derive our Mask R-CNN architecture based on the Functional Pyramid Network (FPN) variant of Mask RCNN.[18] Feature Pyramid Network extracts features at different scales based on their levels in the feature pyramid. The ResNet – FPN backbone provides good accuracy without sacrificing speed.[19]

Our Mask R-CNN architecture leverages and extends the framework of Abdullah et al. at Matterport[20]. Our Mask RCNN model has – 63,744,170 total parameters, 63,632,682 trainable parameters and 111,488 non-trainable parameters.

We observed that the Mask-RCNN requires a large amount of memory and hence we could only train 2 images in a batch per GPU in the available environment reliably without experiencing memory errors. This small batch size was equivalent to stochastic training instead of a mini-batch training. To maintain the network stability we were forced to use a lower learning rate of 0.002 which we had reduce in later epochs to 0.0002.

The results compare with state of the art. We defined two accuracy metrics: accuracy on a "per patch" basis (correctly predicting whether a patch has a given type of tumor, or not) and on an overall image.

Our architecture, pretrained on the COCO dataset, achieved 93% AUC on overall images, and 73% AUC on a per-patch basis.

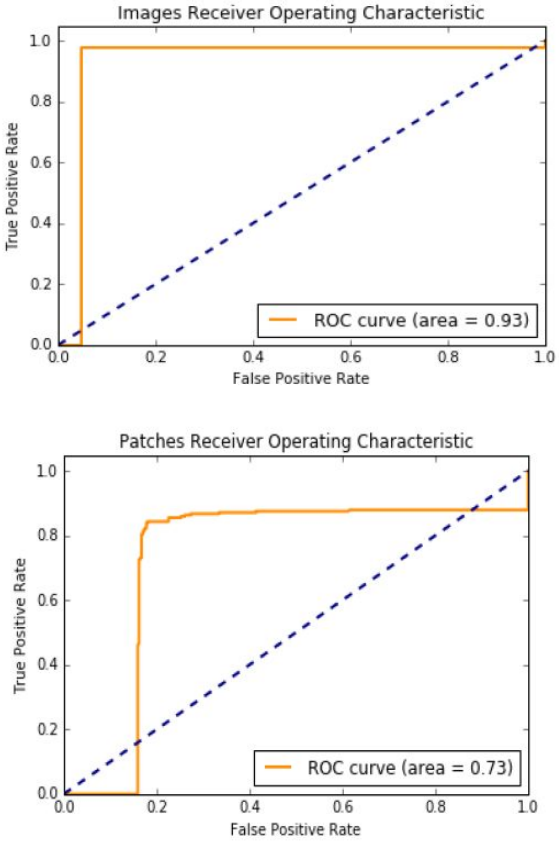


**Fig 3a, 3b: ROC curves on an image and a patch basis**

**Convnets, Patches and Transfer Learning**:

As noted above, the challenge of finding small, fine-grained features in a vast image space indicates that the standard image resizing methods are unlikely to be effective; yet for memory purposes, images that can range up to nearly 5000x4000 in shape must be managed.

One solution is to divide the image into patches by sliding windows, some of which will contain varying pieces of the tumor. We delineated patches of size 256x256 and varied the stride (percent of overlap between images) to generate datasets ranging from 40,000 images to 1,200,000 images, with rotation, flipping, and other image augmentation techniques. We deployed various optimizations (Otsu's method and others) to ensure that the majority of our patches were from the breast rather than the surrounding empty space.

Following training on the patches into four classes (no tumor, malignant, benign, and benign without callback), we tested two validation schemes. First, by splitting each validation image into patches, we declared that if even one patch returned a positive hit for a tumor, the entire image would be flagged as that category. Second, we attempted to use the weights learned by patches as inputs to a final model run on the full images, resized down to 256x256.

Neither result achieved the same level of success as Keras et al [6] and Therapixel [7], likely due to not generating data of sufficient size. Our accuracy (over 4 categories) increased from 45% to 65% by moving from ~200,000 to ~400,000 images, and time and computational resources expired before we were able to attempt a full run on 1.2 million images. Prior authors achieving state of the art results generated 1.2-1.6 million images.

**On Classification Calibration in Neural Networks and Medicine**

Of note, recent work has demonstrated that deep neural networks lose calibration accuracy as they grow more complex, even while increasing accuracy [21]. Calibration specifically refers to the comparison of the model's accuracy with the model's estimated confidence in its predictions. Specifically if a model reports that it is, on average, 80% confident in the images that it classifies as class X, one would expect only 20% of the images in that class to be misclassified. Calibration is of utmost importance, particularly as regards to medicine, where a diagnosing physician would want to trust the model's reported confidence in its prediction.

Guo et al. [21] propose a simple post-processing reweighting scheme to accurately calibrate the model. Of previously cited studies, only Keras et al [6] attempts to estimate a confidence calibration, and do not run any tests to determine if their calibration method is accurate. Further work in this area, particularly for medical imaging, is clearly needed.

**Conclusion and Further Work**

This research project highlighted several aspects of medical image processing. First, due to the high resolution nature of the images, dividing images into patches is absolutely necessary on current image processing hardware.

Second, current standard GAN implementations fail on medical images due to the detail needed, though recent advances suggest promising future approaches.

Third, transfer learning from imagenet does not seem to help with mammogram tumor detection. Future work might investigate whether transfer learning from other types of radiological images (i.e. lung scans) provide more beneficial early weights.

And finally, Mask-RCNN is a fruitful approach to locating and diagnosing tumors. To our knowledge, this is the first such application of this architecture to radiological images.

**References:**

[1] Kinoshita S.K., Marques P.M.A., Slaets A.F.F., Marana H.R.C., Ferrari R.J., Villela R.L. (1998) Detection and Characterization of Mammographic Masses by Artificial Neural Network. In: Karssemeijer N., Thijssen M., Hendriks J., van Erning L. (eds) Digital Mammography. Computational Imaging and Vision, vol 13. Springer, Dordrecht

[2] A. Jalalian, S.B. Mashohor, H.R. Mahmud, M.I. Saripan, A.R. Ramli, B. Karasfi Computer-aided detection/diagnosis of breast cancer in mammography and ultrasound: a review. Clin Imaging, 37 (2013), pp. 420-426

[3] Levy, D. and Jain, A. Breast mass classification from mammograms using deep convolutional neural networks. https://arxiv.org/abs/1612.00542 (2016)

[4] Ertosun MG, Rubin D. Probabilistic visual search for masses within mammography images using deep learning. In: Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference. New York, NY: IEEE; 2015:S1310–S1315.

[5] William Lotter , Greg Sorensen and David Cox: A Multi-Scale CNN and Curriculum Learning Strategy for Mammogram Classification. https://arxiv.org/pdf/1707.06978.pdf (2017)

[6] Geras, K.J., Wolfson, S., Kim, S.G., et al..: High-Resolution Breast Cancer Screening with Multi-View Deep Convolutional Neural Networks. arXiv, https://arxiv.org/pdf/1703.07047 (2017)

[7] Therapixel, Dream Challenge Winning Team Writeup. https://www.synapse.org/#!Synapse:syn9773040/wiki/426908. Accessed December 2, 2017.

[8] Optimizing and Visualizing Deep Learning for Benign/Malignant Classification in Breast Tumors. https://arxiv.org/pdf/1705.06362.pdf

[9] Breast mass classification from Mammograms using Deep Convolutional Neural Networks. https://arxiv.org/pdf/1612.00542.pdf

[10] A multi-view deep learning architecture for classification of breast microcalcifications. http://www.eng.biu.ac.il/goldbej/files/2012/05/paper_isbi_2016.pdf

[11] Deep structured learning for mass segmentation from mammograms. https://arxiv.org/pdf/1410.7454.pdf

[12] Mini Mias Dataset; http://peipa.essex.ac.uk/info/mias.html

[13] CBIS-DDSM https://wiki.cancerimagingarchive.net/display/Public/CBIS-DDSM

[14] Alec Radford, Luke Metz, Soumith Chintala: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

[15] Selective search for object recognition. http://www.cs.cornell.edu/courses/cs7670/2014sp/slides/VisionSeminar14.pdf

[16] ImageNet classification with Deep Convolutional Neural Networks. https://papers.nips.cc/paper/4824-imagenet-classification-with-deepconvolutional-Neural-networks.pdf

[17] Mask R-CNN https://arxiv.org/pdf/1703.06870.pdf

[18] Feature Pyramid Networks for Object Detection. https://arxiv.org/abs/1612.03144

[19] Deep Residual Learning for Image Recognition. https://arxiv.org/abs/1512.03385

[20] Mask R-CNN for object detection and instance segmentation on Keras and Tensorflow. https://github.com/matterport/Mask_RCNN

[21] Guo et al. On Calibration of Modern Neural Networks https://arxiv.org/abs/1706.04599 (2017)

[22] Progressive Growing of GANs for Improved Quality, Stability, and Variation. https://arxiv.org/abs/1710.10196 (2017)