WAYNE STATE UNIVERSITY
COLLEGE OF ENGINEERING

COMPUTER SCIENCE DEPARTMENT

CSC7991: Introduction to Deep Learning

Final Project Report

Applying Deep Learning Methods for Nuclei Detection on Histopathology Images to identify Breast Cancer

Supervised by

Dr. Ming Dong

Submitted by:

Group 7

Debadeep Pharikal
dpharikal@wayne.edu
Access Id – gu5308
Cell# 248-882-5025

Bipasha Banerjee
bipasha@wayne.edu
Access Id – gv1591
Cell# 248-425-1255

Alokparna Bandyopadhyay
bandyopadhyay.alokparna@wayne.edu
Access Id - gq6225
Cell# 248-901-7185

Contents

ABSTRACT

This course project is based on Nuclei Detection on Histopathology Images to identify Breast Cancer using Deep Learning. The major goal of this course project is to experiment deep learning techniques covered under CSC 7991: Introduction to Deep Learning and work towards implementation of those knowledge to develop a near to perfect model in predicting Breast Cancer. The more accurate the models are, more chances of artificial systems to predict if the person is having Breast Cancer. The process of Nuclei detection in high-grade breast cancer images is quite challenging in the case of image processing techniques due to certain heterogeneous characteristics of cancer nuclei such as enlarged and irregularly shaped nuclei. The visual attributes of cells, such as the nuclear morphology, are critical for histopathology image analysis. Based on the proposed cell-level visual representation learning, we further develop a pipeline that exploits the varieties of cellular elements to perform histopathology image classification.

Nuclei Detection on Histopathology Images to identify Breast Cancer

1. Introduction:

1.1 Overview

Breast Cancer is a group of disease in which cells in breast tissue change and divide uncontrollably leading to lump or mass. It is the most common type of cancer which causes 411,000 annual deaths worldwide. After skin cancer, breast cancer is the most common cancer diagnosed in women in the United States and still open area for research especially, the area digital image analysis. Hence the major motivation of this course project is to develop an effective product in medical domain that extracts features from histopathology images and helps in identification of breast cancer.

Supervised learning with convolutional networks (CNNs) has seen huge adoption in computer vision applications but proven to be computationally extensive. Our implementation of this product covers conventional CNN approach to Generative adversarial networks (GAN) for nuclei-detection of histopathology images. GAN models have evolved over time. Thus, our further experiment contains multiple variations of GANs. The results are compared with respect to accuracy and performance, to find best deep learning model for our problem.

1.2 Literature Review

1.2.1 Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) is a special type of artificial neural network architecture with some features of the visual cortex and works phenomenally well on computer vision tasks like image classification, object detection, image recognition, etc. They are composed of many “filters”, which convolve, or slide across the data, and produce an activation at every slide position. But when it comes to data synthesizing and image-to-image translation, CNNs do not work that well. GANs typically works best as image synthesizers and typically use CNN architectures for their generator and discriminator models.

1.2.2 Generative Adversarial Networks (GAN)

Generative adversarial networks (GANs) are algorithmic architectures that use two neural networks, pitting one against the other (thus the “adversarial”) in order to generate new, synthetic or fake instances of data that can pass for real data. The model consists of a generative model and a discriminative model – both realized as multilayer perceptrons (MLPs). In this architecture a generative model G captures the data distribution, and its adversary a discriminative model D estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player.

1.2.3 Conditional Generative Adversarial Networks (CGAN)

The conditional generative adversarial network, or cGAN is a variation of GAN which involves the conditional generation of images by a generator model. Image generation can be conditional on a class label, in case available, allowing the targeted generation of images of a given domain.

1.2.4 Deep Convolutional Generative Adversarial Networks (DCGAN)

Supervised learning with convolutional networks (CNNs) has seen huge success in analyzing histopathological applications. In comparison to that unsupervised learning with CNN has not been successful. The idea of DCGAN help to mitigate that problem. A class of CNNs also known as DCGANs, which have certain architectural constraints, shows that they are suitable for unsupervised learning. We extend this architecture to make it supervised such that it can solve our classification with less computational cost.

1.2.5 Semi Supervised Generative Adversarial Networks (SGAN)

SGAN is an architecture which is based on semi-supervised context by forcing D to output N+1 different output class, N different "real" classes, and an additional fake class (anything that came from G). In our case, N=2 (real nuclei, and real non-nuclei).

1.2.6 Least Squares Generative Adversarial Networks (LSGAN)

Regular GANs hypothesize the discriminator as a classifier with the sigmoid cross entropy loss function. This loss function may lead to the vanishing gradients problem during the learning process. To overcome such a problem, the conventional loss function for discriminator is replaced with least squares loss function. There are two benefits of LSGANs over regular GANs. First, LSGANs can generate higher quality images than regular GANs. Second, LSGANs perform more stable during the learning process.

1.2.7 Coupled Generative Adversarial Networks (CoGAN)

COGAN consists of pair of GANs. Ming et al proposed CoGAN architecture can learn a joint distribution without any tuple of corresponding images. It can learn a joint distribution with just samples drawn from the marginal distributions. This is achieved by enforcing a weight-sharing constraint that limits the network capacity and favors a joint distribution solution over a product of marginal distributions one.

1.3 Objective & Contribution

The major motivation of this course project is to develop an effective product using Deep Learning method especially in the field of GANs and provide the comparable accuracy of the literature study done. Our novel contribution is to build all GAN models using supervised or semi-supervised learning methods and to choose best GAN model for classification of histopathology images in predicting breast cancer. Additionally, we will change some of the architectural constraints to fine tune the model accuracy and make the model suitable for supervised/semi-supervised learning. We propose additional momentum $\beta 2$ which stabilized most of our training procedure.

We worked as a team of 3 and task load was distributed equally and took paired-programming approach. Challenges were fixed as combined team-effort. Just for division purpose we picked 2-3 models per person. The results were thoroughly analyzed and tested before delivering the final product.

- CNN, LSGAN – Contributed by Alokparna Bandyopadhyay

- GAN, CGAN, DCGAN – Contributed by Bipasha Banerjee
- SGAN, COGAN – Contributed by Debadeep Pharikal

2. Experimental Set-up & Procedure

2.1 Dataset

The original data set contains 537 hematoxylin–eosin (H & E) stained histopathological images obtained from digitized glass slides. H&E stained breast histopathology glass slides were scanned into a computer using a high-resolution whole slide scanner, Aperio ScanScope digitizer, at 40x optical magnification.

The dataset corresponds to 49 lymph node-negative and estrogen receptor positive breast cancer (LN-, ER+ BC) patients at **Case Western Reserve University**. For sake of running the models with our existing configurations we borrowed the dataset used by Xu et al which contains less amount samples.

a) The training data includes 2,000 nuclear and 6,000 non-nuclear patches.
b) There are 1,000 patches for validation, 500 nuclear patches and 500 non-nuclear.
c) This dataset already contains the data divided into train and validation. The value range of each image was originally [0…1]. but we normalize it to be [-1…1]. We modify training and testing labels. 0 represents non-nucleus, 1 represents nucleus.

The dataset can be obtained from link: https://engineering.case.edu/centers/ccipd/data

2.2 Parameter Setting

- Image from the training dataset was downsized to $32 \times 32 = 1024$ pixels which was originally 34 X 34. We performed this downsizing as it is enough to contain a nucleus within the patch under 40X optical magnification resolution.
- No of Classes = 2 (Nuclei & Non-Nuclei)
- Each patch size has three color channels ($\tau = 3$). Therefore, there are $d_x = s_0 = 32 \times 32 \times 3 = 1024 \times 3$ input units in the input layer.
- We used Adam optimizer with tuned hyperparameters with 0.0002 as the learning rate, 0.5 as momentum variable β_1 and 0.1 as momentum variable β_2, which helped stabilize training. We have borrowed the learning rate and momentum parameter as suggested by Xu et al, however additionally used another momentum term β_2 which helped to increase the training accuracy compared to the mentioned literatures for some GAN models.
- Loss parameters used are *Binary cross-entropy* and *categorical entropy* for all GAN models except *LSGAN* where Binary cross-entropy is replaced with *mean squared error (MSE)*. Binary cross-entropy is used to distinguish among real or fake samples and categorical entropy is to distinguish among which real category is (nuclei or non-nuclei).

2.3 Training Procedure

The training procedure is almost same for all the models and modified as per architectural needs.

- The CNN model was trained for 20 epochs, with a batch size of 128
- All the GAN models were trained for 200 epochs, with a batch size of 32
- All the GAN models (except SGAN) are made supervised which is also an innovation part

of our project

- In SGAN, for every epoch, we selected a 50% of half batch size of images (16 images) from the TMI training set and another random half samples from a Gaussian distribution.

2.3.1 Training CNN for Nuclei Detection

The 2-CNN model used in this research inherits from the Keras Model and is used to classify whether an input histopathology image has a Nuclei or not.

- ***CNN Architecture***
  The first layer is a 2D convolution layer with 128 filters and 'relu' activation function, followed by a 2D max-pooling layer. The third layer consists of another 2D convolution layer with 256 filters and 'relu' activation function, followed by another 2D max-pooling layer. We have used two fully connected dense neural network layers with 'relu' activation functions. The output layer consists of a third dense layer with 'sigmoid' activation function, which determines whether an in image belongs to the 'Nuclei' class or to the 'Non-nuclei' class.
- ***Training the CNN Model***
  The CNN model is compiled using 'categorical cross-entropy' loss for image classification and 'Adam' optimizer with learning rate of 0.0002, momentum as beta_1=0.5 and beta_2=0.1. The model is then trained for 20 epochs and 128 batch size.

  Training time: 72.2 minutes (for epoch size of just 20)

2.3.2 Training GAN for Nuclei Detection

The training procedure of GAN is naïve, and we followed the architecture provided in the literature and modified them as per our need.

- ***Training the Generator (G)***
  The first layer of Generator (G) is Dense and a size of $128x8x8$ is instantiate with ReLU activation function. After a series of Dense, leaky RELU, batch normalization, , we further instantiate the model outputs a $32x32x3$ RGB tensor shape that is normalized between values of [-1…1] through the Hyperbolic Tangent Function (tanh) and finally the image is reshaped.
- ***Training the Discriminator (D)***
  After a series of convolutions, batch normalization, leaky RELUs and dropout, model is instantiated with two activation functions. A sigmoid activation function is used to indicate the predicted probability of the given image being real or fake, and, if and only if the image is real, a softmax activation function with $K = 2$ classes to indicate the predicted probabilities of the given image being nucleus (label=1) or non-nucleus (label=0).

  Training time: 2.5 minutes

2.3.3 Training CGAN for Nuclei Detection

- The procedure for training CGAN is similar to GAN.
- Only in case of input for both generator and discriminator a conditional variable is provided which acts as a switch to generate image in different domain.

Training time: 1.9 minutes

2.3.4 Training DCGAN for Nuclei Detection

- ***Training the Generator (G)***
  The first layer of Generator (G) is Dense and size of 8×8×128. Next our approach consisted of reshaping a random vector z to have a 4D shape and then pass it to a sequence of transpose convolutions, batch normalization and RELU operations that increase the spatial dimensions of the input vector while decreases the number of channels. As a result, the network outputs a 32x32x3 RGB tensor shape that is normalized between values of [-1…1] through the Hyperbolic Tangent Function (tanh).
- ***Training the Discriminator (D)***
  The random half batch of images from a Gaussian distribution when given to generator, it transforms this noise into a set of fake images. To stabilize the discriminator, we used learning process with mini-batch discrimination. After a series of convolutions, batch normalization, leaky RELUs and dropout, the model is instantiated with sigmoid activation functions to determine if the image is real or fake. If and only if the image is real, a softmax activation function with K=2 classes to indicate the predicted probabilities of the given image being nucleus (label=1) or non-nucleus (label=0).

Training Time:6.4 minutes

2.3.5 Training SGAN for Nuclei Detection

- Same architecture from DCGAN are used for generator and discriminator to train a semi supervised model.
- For each epoch, we selected a random half batch of images (16 images) from the TMI training set and another random half samples from a Gaussian distribution.
- To balance the difference in occurrences of class labels, 50% of labels that discriminator trained on are "fake", i.e., class weights were divided equally. This approach is called Mini-batch discrimination.
- Class weight = 0.5

Training Time: 27.4 minutes

2.3.6 Training LSGAN for Nuclei Detection

- The Generator architecture and training procedure for LSGAN is similar to GAN.
- ***Training the Discriminator (D)***
  Adopted the Mean Square Error (MSE) loss function for training the discriminator, which is a simple three layer dense fully connected neural network. A sigmoid activation function

is used to indicate the predicted probability of the given image being real or fake, and, if and only if the image is real, a softmax activation function with $K = 2$ classes to indicate the predicted probabilities of the given image being nucleus (label=1) or non-nucleus (label=0)

Training time: 1.1 minutes

2.3.7 Training CoGAN for Nuclei Detection

- We partitioned the TMI training set into two disjoint subsets.
- We applied a 90-degree rotation to all the images in the second set to construct the second domain. There were no corresponding images in the two domains.
- ***Training the Generator (G)***

  Weight Sharing Layers
  - The first layer of Generator (G) is Dense and size of 8×8×128 followed by LeakyReLU and Batch normalization with momentum of 0.8. Series of dense, LeakyReLU and Batch Normalization layers followed next.
  - Hence ensured same high-level features being generated from both generators G1 and G2.

  Without Weight Sharing Layers

  The layer is different and passed through a Dense, LeakyReLU, BatchNormalization and instantiated further with Hyperbolic Tangent Function (tanh) and then reshaped.

- ***Training the Discriminator (D)***

  Weight Sharing Layers
  - The image is Flattened and passed through couple of Dense Layers followed by LeakyReLU.
  - Further the image is instantiated with a sigmoid activation function to determine if the image is real or fake.

  As weight sharing has no impact on discriminator, but to minimize the training parameters for discriminator no separate layer is being used for D1 & D2.

3. Experiment Results

The performance result of all the GAN model and CNN will be compared and chosen which deep learning model best suits for classifying Nuclei Vs Non-Nuclei. For all our GAN models we did not use the TMI testing set. We used the training set; it was already pre-divided into 8,000 samples for training and 1,000 samples for validation. We used testing dataset and computed the standard classification definition of Precision, Recall or True Positive Rate (TPR), False Positive Rate (FPR), F-measure, and Average Precision (AveP) over the testing dataset with 1000 samples.

Evaluation Metrics

The evaluation of each metric is given by.

$Precision = \frac{TP}{TP + FP}$	$Recall = \frac{TP}{TP + FN}$	$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$	$AveP = \int_0^1 p(r)dr$

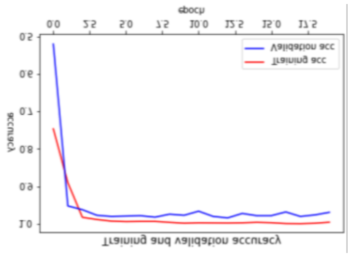
where,
$TP + TN$ = total number of nuclei patches
$FP + TN$ = total number of non-nuclei patches
$AveP$ = average area under Precision-Recall curve.

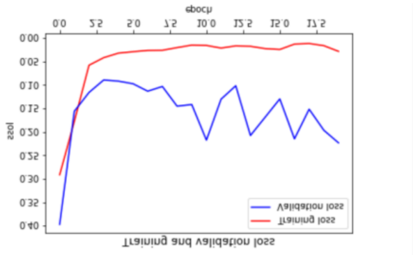
CNN

	Precision (%)	*Recall (%)*	*F-measure (%)*	*Support (%)*
Non-Nucleus	95	98	97	500
Nucleus	99	95	97	500

Overall accuracy: 96.70%

Overall Loss: 22.14%



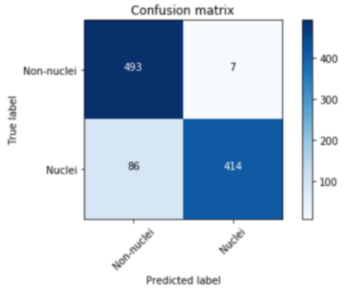


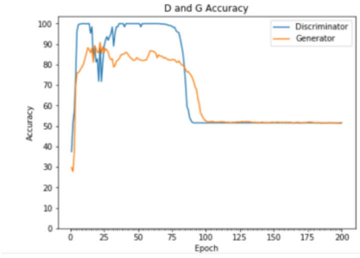
GAN

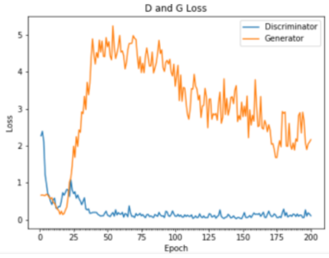
	Precision (%)	*Recall (%)*	*F-measure (%)*	*Support (%)*
Non-Nucleus	85	99	91	500
Nucleus	98	83	90	500

Overall accuracy: 90.700000%

AveP: 90.023278%





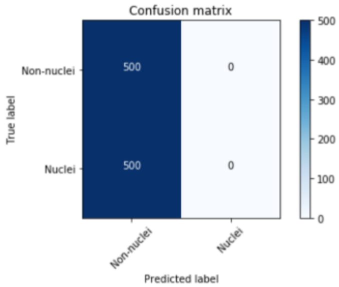


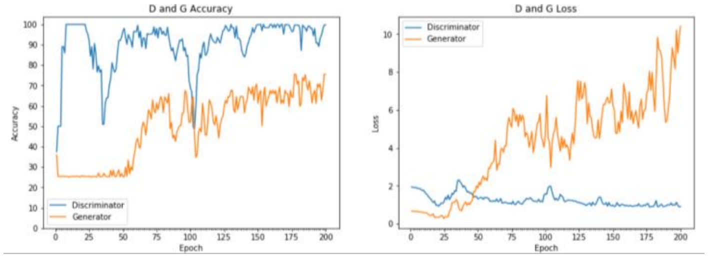
CGAN

	Precision (%)	***Recall (%)***	***F-measure (%)***	***Support (%)***
Non-Nucleus	50	100	67	500
Nucleus	0	0	0	500

Overall accuracy: 50.000000%

AveP: 50.000000%



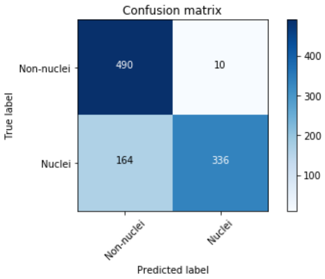


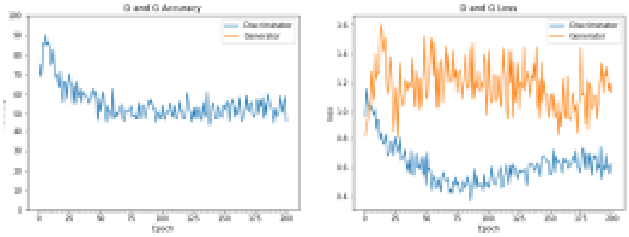
DCGAN

	Precision (%)	***Recall (%)***	***F-measure (%)***	***Support (%)***
Non-Nucleus	75	98	85	500
Nucleus	97	67	79	500

Overall accuracy: 82.600000%

AveP: 81.657803%



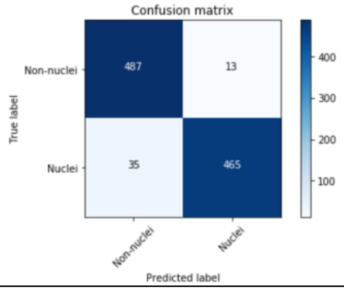


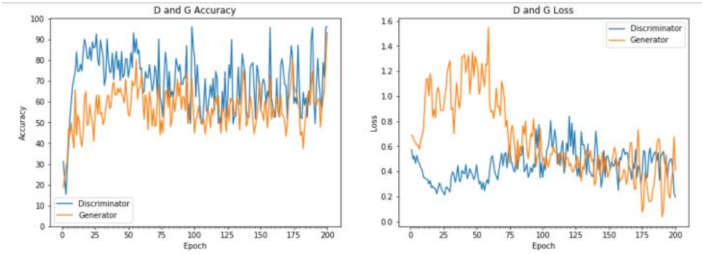
SGAN

	Precision (%)	***Recall (%)***	***F-measure (%)***	***Support (%)***
Non-Nucleus	93	97	95	500
Nucleus	97	93	95	500

Overall accuracy: 95.200000%

AveP: 93.970711%



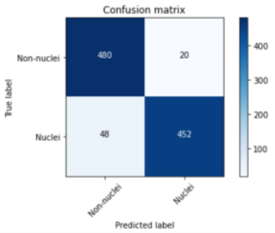


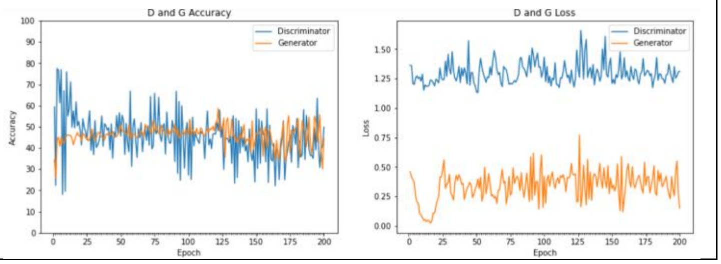
LSGAN

	Precision (%)	***Recall (%)***	***F-measure (%)***	***Support (%)***
Non-Nucleus	91	96	93	500
Nucleus	96	90	93	500

Overall accuracy: 93.200000%

AveP: 91.369492%



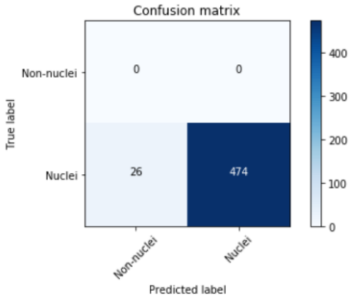


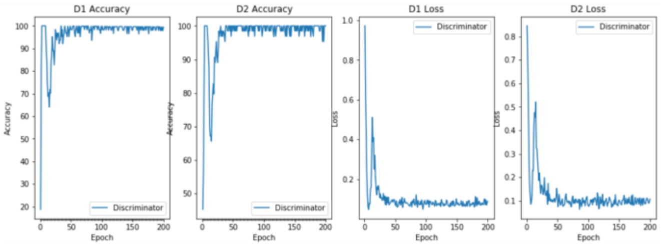
CoGAN

	Precision (%)	***Recall (%)***	***F-measure (%)***	***Support (%)***
Non-Nucleus	85	99	91	500
Nucleus	98	83	90	500

Overall accuracy: 97.000000%

AveP: 96.083268%





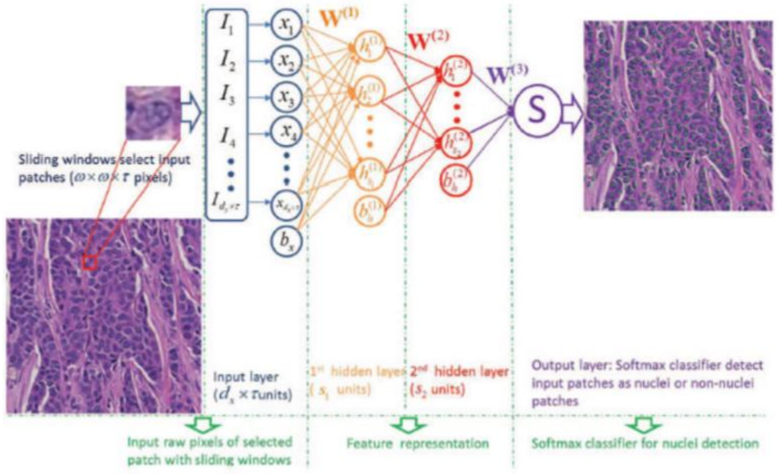
Comparison

	Overall Accuracy	Average Precision	Average Recall	F-Measure	Execution Time (in mins)
CNN	96.7	97	97	97	72.2(**for 20 epochs**)
GAN	90.7	91.5	91	90.5	2.4
CGAN	50.0	25.0	50.0	33.5	0.3
DCGAN	77.7	84.5	78.25	76.5	6.4
SGAN	95.2	95	95	95	27.4
LSGAN	93.2	93.5	93	93	1.1
CoGAN	97.1	99	94	97	6.3

Overall Summary

We observed CNN performed well as its supervised classification. However, we will proceed comparison with only GAN models as we discarded CNN by running after 20 epochs. With the hardware configuration used while developing the product, the memory usage was becoming extensively high and an epoch of 200 can take entire day to compute. We provide our comparable results only with GAN models as it stands superior in comparison w.r.t computation power & execution time.

- For all our GAN models we worked as if a sliding window detector is first enacted to randomly select image patches before feeding to the model.
- Then high-level features are extracted via this model and this feature are then subsequently input to SMC. Finally, the trained SMC classifies each image patch as either having or not having a nucleus present.



Sliding windows select input patches ($\varpi \times \varpi \times \tau$ pixels)
I_1
I_2
I_3
I_4
$I_{d_x \times \tau}$
x_1
x_2
x_3
x_4
$x_{d_x \times \tau}$
b_x
$\mathbf{W}^{(1)}$
$h_1^{(1)}$
$h_2^{(1)}$
$h_{s_1}^{(1)}$
$b_h^{(1)}$
$\mathbf{W}^{(2)}$
$h_1^{(2)}$
$h_{s_2}^{(2)}$
$b_h^{(2)}$
$\mathbf{W}^{(3)}$
S
Input layer ($d_x \times \tau$units)
1st hidden layer (s_1 units)
2nd hidden layer (s_2 units)
Output layer: Softmax classifier detect input patches as nuclei or non-nuclei patches
Input raw pixels of selected patch with sliding windows
Feature representation
Softmax classifier for nuclei detection

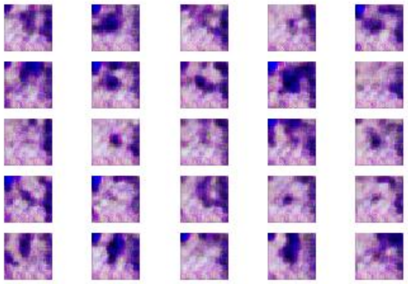
- Vanilla GAN or simple GAN performance is also impressive and most important factor is the execution time. It proves why GAN models are comparatively better than traditional CNNs.
- CGAN performed poorest, however we believe that the accuracy can be improved by better image processing. We could not work further owing to time constraint. Hence this will be future scope of the project. We can explore more on one side label smoothing, Virtual Batch Normalization (VBN), and Inception Scoring.
- We see exceptionally high results for Co-GAN. It even demonstrates 98% as precision for the discriminator to label as nucleus which is exceptionally well. However, there can be a chance of over fitting with so much of accuracy. Scope for future analysis.
- SGAN is the only semi-supervised technique experimented. **We must keep in mind, to solve a typical problem like this where label data is often difficult and expensive to obtain, semi-supervised learning method is always good approach.** This semi-supervised model, even computationally costlier than other supervised model still gave a

good metric with respect to precision and recall found even better than literature referred by *V.M Vargas et al*.

Model	***Precision (%)***	***Recall (%)***	***F-measure (%)***	***AveP (%)***
SGAN(as per literature)	**94**	**94**	**94**	**93**
SGAN (modified by us)	**95**	**95**	**95**	**94**

- According to us, LSGAN gave optimal result for a supervised learning task. A precision of 96% is intuitively a good ability of discriminator, D to not to label as nucleus a sample that is not a nucleus. A recall of 93% is intuitively a good ability of discriminator, D to find all the nucleus samples.

Model Output:



Source code and all model output and processed images can be accessed from the GitHub Link: https://github.com/pharikal/Breast-Cancer-Detection-Using-Deep-Learning

4. Conclusion

This course project explores, and reviews various deep learning techniques used for histopathology image analysis with a goal on breast cancer detection. We compared multiple GAN models and showed how an efficient deep-learning model can capture high-level feature representations of pixel intensity in a supervised and semi-supervised manner. LSGAN stands out the best for a supervised learning task owing to its high AveP and SGAN comes handy when we have less labelled data compared to Supervised GAN models. So, we conclude that these high-level features enable the classifier to work very efficiently for detecting multiple nuclei from a large cohort of histopathological images as well as to generate realistic synthesized representations of nuclei and non-nuclei images. This review aims at complementing the effort of pathologists, in examining and analyzing biopsy samples, by computer aided techniques and thereby help medicine and science to predict breast cancer.

Acknowledgement

We thank professor Dr. Ming Dong for his guidance with the concepts of Deep Learning & his insights regarding the approach of solving a classification task like this and teaching assistant Mr. Qisheng He for his valuable inputs with Keras & Tensorflow.

References

1. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, & Y. Bengio. (2014, Jun.). "Generative Adversarial Networks." ArXiv E-Prints. [On-line]. Available: https://arxiv.org/abs/1406.2661 [7 May 2018].
2. A. Radford, L. Metz, & S. Chintala. (2016, Jan.). "Unsupervised representation learning with deep convolutional generative adversarial networks." ArXiv E-Prints. [On-line]. Available: https://arxiv.org/abs/1511.06434 [7 Jan 2016].
3. Unsupervised Learning for Cell-level Visual Representation in Histopathology Images with Generative Adversarial Networks Bo Hu] , Ye Tang] , Eric I-Chao Chang, Yubo Fan, Maode Lai and Yan Xu*. https://arxiv.org/pdf/1711.11317.pdf
4. V. Vargas & J. Koller. "GAN-for-Nuclei-Detection." Internet: https://github.com/vmvargas/GAN-for-Nuclei-Detection , May 7, 2018 [7 May 2018].
5. J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang, & A. Madabhushi, "Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images," in IEEE Transactions on Medical Imaging, Vol. 35 no. 1, pp. 119-130, July 2016. https://europepmc.org/article/pmc/pmc4729702
6. Keras implementation of General Adversarial Network - https://github.com/eriklindernoren/Keras-GAN
7. Coupled GAN code by authors - https://github.com/mingyuliutw/CoGAN
8. Conditional GAN - https://machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch/