Semi-Supervised Generative Adversarial Network (SGAN) for Nuclei Detection on Breast Cancer Histopathology Images

Victor M. Vargas
Seidenberg School of CSIS, Pace University.
New York City, NY.
victor@vmvargas.com

Jonathan Koller
Seidenberg School of CSIS, Pace University.
New York City, NY

***Abstract*—This paper refine the work of a Semi-Supervised Generative Adversarial Network (SGAN) to work with the H&E breast cancer histopathology images dataset published by the Case Western Reserve University. Our goal was to evaluate if the semi-supervised model could achieve comparable or better performance to state-of-the-art approaches in nuclei detection. Our results showed that our model was able to learn useful high-level features of nuclear structures as well as to generate visually appealing samples. We conclude that more experimentation is necessary in order to formulate a more robust and significant progression.**

Keywords— Feature representation learning; automated nuclei detection; semi-supervised approach; Generative Adversarial Network; breast cancer histopathology.

I. Introduction

Nuclei Detection allows researchers to identify each individual cell in a sample. By measuring how cells react to various treatments, the researcher can understand the underlying biological processes at work. The analysis of histopathology images is currently the standard for diagnosing Breast Cancer (BC). This fact is a convincing motivation to discover, enhance, and automated efficient approaches to distinguish individual cancer nuclei on breast pathology images.

Getting large amounts of unlabeled medical data is generally much easier than labeled data. Unsupervised generative models with stochastic components such as Generative Adversarial Networks (GAN) and Variational Autoencoder (VAE) can be trained end-to-end to learn representative features in a completely unsupervised way. For that reason, both approaches could optimally leverage this amount of information.

The following is an outline of the rest of this paper. A review of similar architectures and previous related works is presented in Section II, a detailed description of our Semi-Supervised Generative Adversarial Network (SGAN) is presented in Section III, the experimental setup and comparative strategies are discussed in Section IV, the experiment results and discussions are reported in Section V, and conclusions and future work are presented in Section VI.

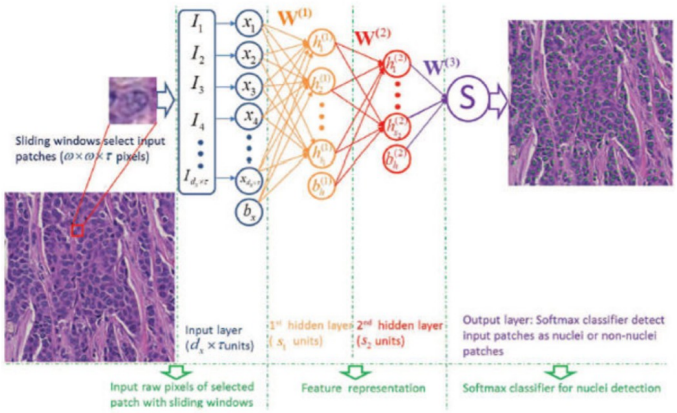

Figure 1. Illustration of SSAE+SMC for nuclei detection on breast histopathology

II. Previous Related Work

A. Stacked Sparse Autoencoder (SSAE) for Nuclei Detection on Breast Cancer Histopathology Images

An interesting approach, especially in cases where object annotation to generate training data is expensive, are the integration of multiple instance learning (MIL) and deep learning. Xu et al. [1] investigated the use of a MIL-framework with both supervised and unsupervised feature learning approaches as well as handcrafted features. The results demonstrated that the performance of the MIL-framework was superior to handcrafted features, which in turn closely approaches the performance of a fully supervised method.

Training an SSAE+SMC model (SMC stands for Softmax Classifier), involves finding the optimal parameters $\theta=\left(W, b_h, b_x\right)$ simultaneously by minimizing the discrepancy between input and its reconstruction. After the optimal parameters θ are obtained, the SSAE+SMC yields a function that transforms input pixel intensities of an image patch to a new feature representation of nuclear structures.

As Fig. 1 shows, with an SSAE+SMC model, each training patch $x(k)$ of pixel intensities is represented by a high-level structured representation of nuclei or non-nuclei patches $h^{(2)}(k)$ in the second hidden layer of the model. Note that in the SSAE

learning procedure, the label information y is not used. Hence, SSAE learning is an unsupervised learning scheme.

During detection process, each image patch detected by a sliding window is first represented by high-level feature $h^{(2)}(k)$. This is then fed to the SMC and produces a value between 0 and 1 that can be interpreted as the probability of the input image patch corresponding to a nucleus or not.

Similarly to the SSAE model, our model (G) can transform the input pixel intensities to structured nuclei or non-nuclei representations. Therefore, our GAN based framework is also able to learn high-level structure information from a large number of unlabeled image patches. Both approaches (SGANs and VAEs) are generative models. However, even though VAEs tend to have a clearer and objective cost function, we decided to use an SGAN framework because: (1) unlike VAE, GANs are able to generate very realistic samples and (2) recently published papers revealed effective techniques to produce more stable GANs [2][3].

B. Generative Adversarial Networks (GAN)

GANs are based on a game-theoretic scenario in which the generator network must compete against an adversary. The generator network (G) directly produces samples $x = g(z; \theta(g))$. Its adversary, the discriminator network (D), attempts to distinguish between samples drawn from the training data and samples drawn from the generator. The discriminator emits a probability value given by $d(x; \theta(d))$, indicating the probability that x is a real training example rather than a fake sample drawn from the model [4].

The representations that can be learned by a GAN may be used in a variety of applications, including image synthesis, semantic image editing, style transfer, image super-resolution, and classification [5]. Expanding these ideas, one can produce good output samples using a set of convolutional neural networks [6]. Some years ago, Radford, Metz, & Chintala [2] created surprisingly good samples from a single generator network.

C. DCGAN

Several recent papers have focused on improving the stability of training and the resulting perceptual quality of GAN samples [6][7][8][9]. Among these, the main contribution by Radford et al. [2] came from a set of practices that prove to stabilize the training of GAN by: (1) replacing deterministic spatial pooling functions (i.e. max-pooling) with strided convolutions, (2) eliminating fully connected layers on top of convolutional features, and (3) not applying Batch Normalization to the generator output layer and the discriminator input layer [10].

We use some of these architectural innovations proposed by Radford et al. as discussed in Section III.

III. SGAN

Odena proposed an extension of the DCGAN architecture to the semi-supervised context by forcing D to output N+1 different output classes, N different "real" classes, and an additional fake class (anything that came from G) (2016). In our case, N=2 (real nuclei, and real non-nuclei).

Using generative models on semi-supervised learning tasks is not a new approach. Kingma & Welling [11] expand the work on variational generative techniques [12][13] to do just that. However, Odena [14] described a new extension called SGAN that improves classification performance on restricted data sets over a baseline classifier with no generative component (2016).

According to Odena, training an SGAN is similar to training a GAN. One simply use higher granularity labels for the half of the minibatch that has been drawn from the data generating distribution. D is trained to minimize the negative log likelihood with respect to the given labels and G is trained to maximize it, as shown in Algorithm 1.

Algorithm 1 SGAN Training

Input: I: number of total iterations

for $i = 1$ **to** I **do**

1: Draw m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.

2: Draw m examples $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ from data generating distribution $p_d(x)$.

3: Perform gradient descent on the parameters of D w.r.t. the NLL of D's outputs on the combined minibatch of size $2 \times m$.

4: Draw m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.

5: Perform gradient descent on the parameters of G w.r.t. the NLL of D's outputs on the minibatch of size m.

end for

Our work can be seen as a refinement of this method adapted to perform nuclei detection.

IV. Experimental Setup

At first, we conducted experiments on MNIST to evaluate whether the classifier component (D) of our model could perform as intended and similar to the original SGAN implementation. Then, we conducted experiments on the TMI Dataset (the Breast Cancer histopathology images).

The experiments in this paper were conducted by Vargas & Koller at Pace University [15]. This work borrowed heavily from Linder-Norén and contains more details about the experimental setup [16].

D. TMI Dataset

To be able to compare the experimental results with the SSAE model, the experimental setup for the dataset was almost identical to Xu et al. [1]. We used the same dataset of 537 H&E stained histopathological images, obtained from digitized glass slides corresponding to 49 lymph node-negative and estrogen receptor-positive breast cancer (LN-, ER+ BC)

patients at Case Western Reserve University. The training data includes 2,000 nuclear and 6,000 non-nuclear patches. There are 1,000 patches for validation, 500 nuclear patches and 500 non-nuclear. Xu et al. explained the generation of the training and ground truth datasets. In this paper, we did not use the testing dataset. Instead we used the validation as the testing dataset.

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.

E. Parameter setting

To train the model with the TMI dataset, the patch size was initially defined as 34×34 pixels. However, due to fact that G input image has to start from an integer number, every image from the training dataset was downscaled to $32 \times 32 = 1024$ pixels which is big enough to contain a nucleus within the patch under 40X optical magnification resolution images. 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.

Regarding the optimizer, we used the same Adam optimizer cited and then replicated by Odena [14][17]. We use 0.0002 as the learning rate, and 0.5 as momentum term $\beta 1$, which helped stabilize training. Both, G and D used the same optimizer. Figure 2 and Figure 3 depicts the architecture of both models, which is very similar to the DCGAN.

We used almost all architecture guidelines suggested by Radford et al. [2] for a stable Deep Convolutional network (2015):

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

A notably difference in our model relies on G first layer which is an $8 \times 8 \times 128$ this size assures that G outputs an image with equal size as the training images.

F. Training the SGAN for Nuclei Detection

The training procedure was for 200 epochs, with a batch size of 32 images. The number of epochs was chosen after evaluating our model training history in terms of accuracy and loss (see Fig. 5). We find that the highest accuracy and lowest loss for both (G and D) was reached around the 185 to 205 epochs of training.

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 D trained on are "fake", i.e., class weights were divided equally. This approach is called Mini-batch discrimination [18]. SGAN average training time was 6 minutes and 12 seconds using Keras, Tensorflow in the backend on a GPU Nvidia GeForce GT 755M.

1) Training the Discriminator (D)

The random half batch of images from a Gaussian distribution were given to G who transform this noise into a set of synthetic images. The goal was to stabilize D learning process with mini-batch discrimination.

D, now a multi-class classifier, is the most relevant network for this architecture. After a series of convolutions, batch normalization, leaky RELUs and dropout, we instantiate the model with two activation functions. A sigmoid activation function 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).

D is trained by minimizing the cross-entropy between the observed labels and the model predictive distribution $p_{model}(y \vee x)$.

2) Semi-supervised learning for Nuclei Detection

The way D does semi-supervised learning is by adding samples from G to D's data set, labeling them with a new "generated" class $y = K+1$, and correspondingly increasing the dimension of D's output from $K = 2$ to $K = 2+1$. We then use $p_{model}(y = 2+1 | x$ to supply the probability that a given image x is fake, i.e., coming from G. With this in mind, we could learned from unlabeled data, as we knew that it corresponded to one of the K classes of nuclei data by maximizing $\log p_{model}(y \in \{0, \ldots, K\} \vee x)$.

3) Training the Generator (G)

G followed a very standard implementation described in the DCGAN paper. Our approach consisted of reshaping a random vector z to have a 4D shape and then feed it to a sequence of transpose convolutions, batch normalization and leaky RELU operations that increase the spatial dimensions of the input vector while decreases the number of channels. As a result, the network outputs a $32 x 32 x 3$ RGB tensor shape that is normalized between values of [-1…1] through the Hyperbolic Tangent Function (tanh).

G is trained 10 times per epoch to overcome a failure mode in where D overpowered G, classifying generated images as fake with absolute certainty and leaving no gradient for the generator to descend. The number 10 was empirically defined. Fig. 4 shows a randomly generated batch of 25 images per epochs during G's training phase. Fig. 5 shows D and G accuracy and loss per epoch.

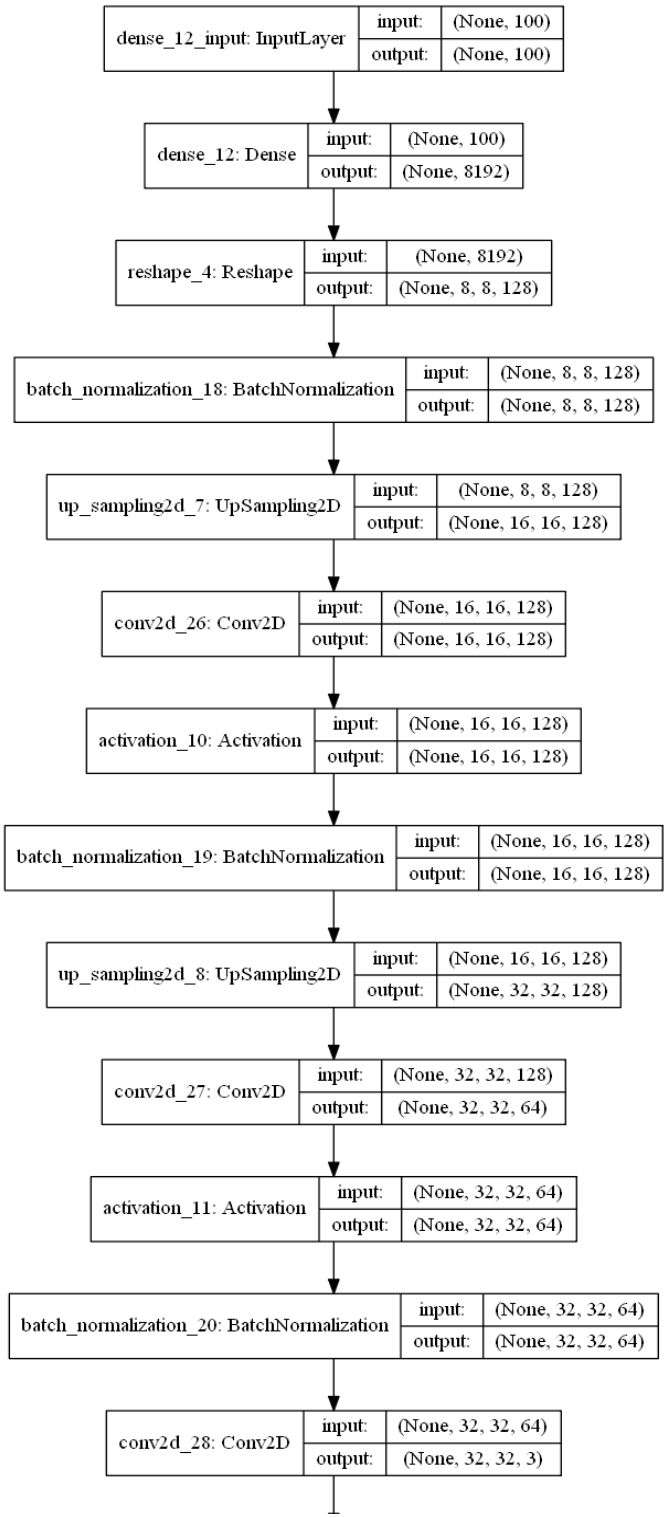

Figure 2. G model's graph.

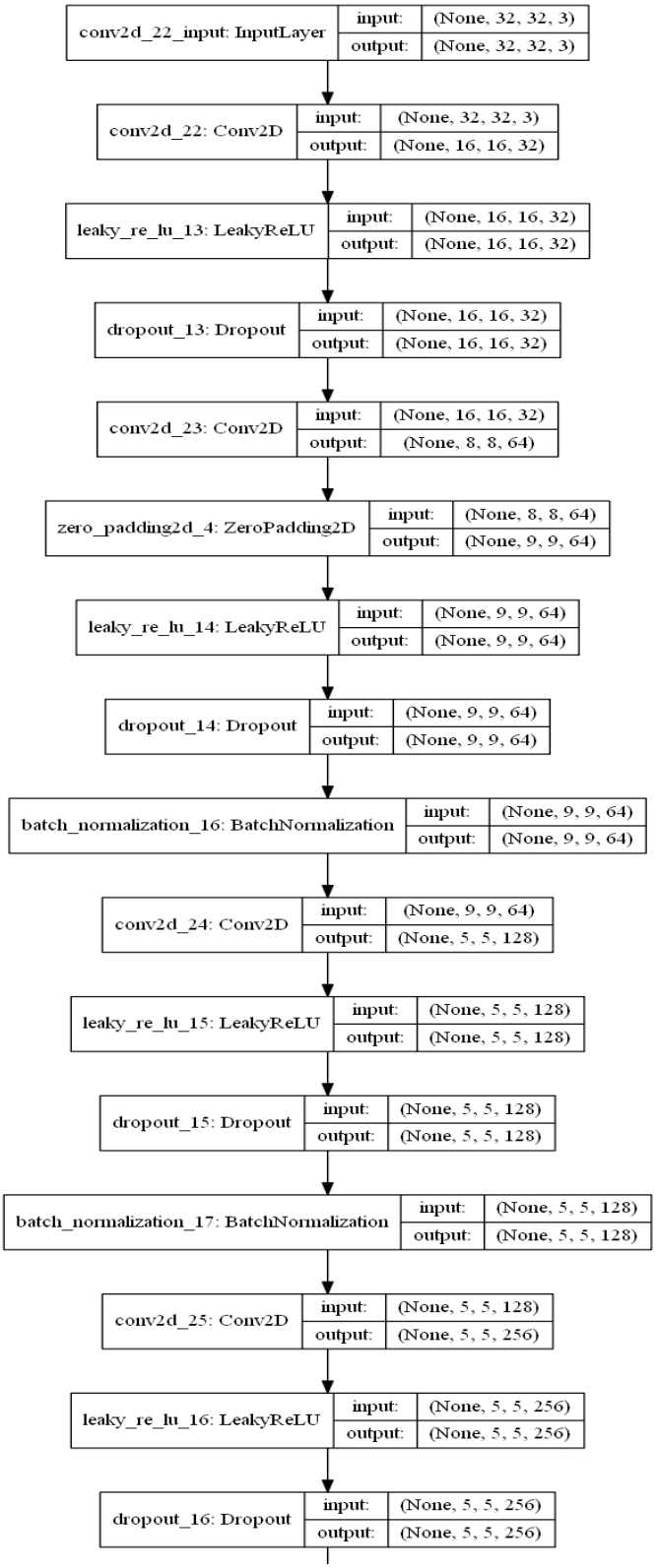

Figure 3. D model's graph.

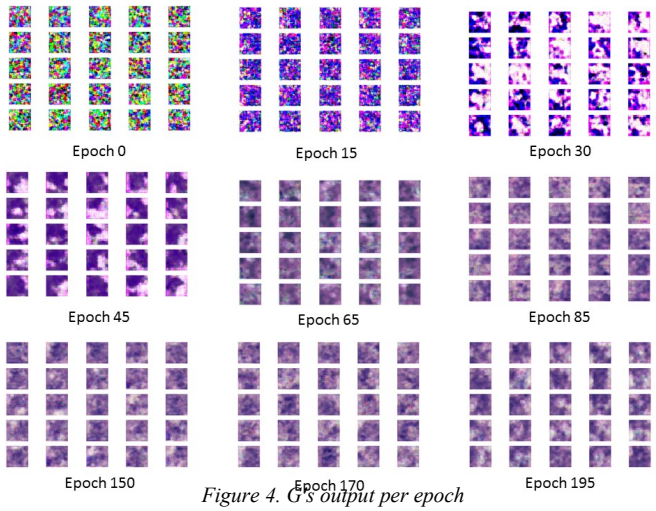

Figure 4. G's output per epoch

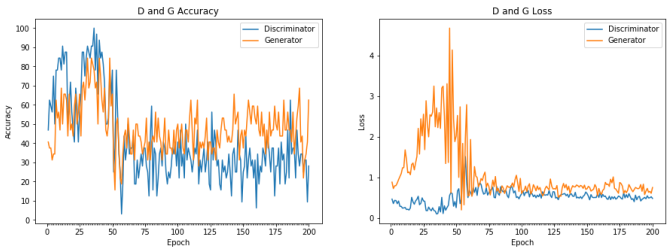

Figure 5. D and G accuracy and loss per epoch

G. Experimental design and comparative strategy

In order to show the effectiveness of our model for the problem of nuclei detection, the SGAN model was compared against the state-of-the-art model Stacked Sparse Autoencoder plus Softmax Classifier (SSAE+SMC) [1].

For SSAE, the detection procedure is the same as illustrated in Figure 1. A sliding window detector is first employed to select image patches before feeding to the model. Then high-level features are extracted via this model and this features are then subsequently input to SMC. Finally, the trained SMC classifies each image patch as either having or not having a nucleus present.

H. Performance Evaluation

The performance of automatic nuclei detection was quantified in terms of Precision, Recall or True Positive Rate (TPR), False Positive Rate (FPR), F-measure, and Average Precision (AveP).

$$Precision = \frac{TP}{TP + FP}$$

$$Recall \vee TPR = \frac{TP}{TP + FN}$$

$$Fmeasure = 2 \times \frac{PrecisionxRecall}{Precision + Recall}$$

$$AveP = \int_0^1 p(r)\,dr$$

However, the definition of each term is different for each approach. Xu et al. used the testing dataset [1]:

a) True Positive (TP)

Defined as the number of nuclei correctly identified as such by SSAE model. In the paper by Xu et al., the correct detection of nuclear patches (true positives) was identified as those instances in which the distance between the center of the detected nuclear window and the closest annotated pathologist identified nucleus was less than or equal to 17 pixels.

4) TP + FN

Defined as the total number of automatically detected nuclei. FP+TN is defined as the total number of patches without nuclei.

5) Average Precision (AveP)

Involves computing the average value of $p(r)$ over interval between $r = 0$ and $r = 1$ and the precision $p(r)$ is a function of recall r. Therefore, AveP shows the average area under Precision-Recall curve.

For our model (SGAN), we did not used the TMI testing set. Instead, we used the entire training set, it was already divided into 8,000 samples for training and 1,000 samples for validation. We used the 1,000 as 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 1,000 testing dataset.

V. Experimental Results

D's overall classification accuracy across the 1,000 testing set was 94.10%. Thus, D's error rate is 0.59. Figure 6 and 7 shows the confusion matrices to visualize the performance of our SGAN model. It is worth to mention that the system makes a highly accurate distinction between nucleus and non-nucleus, as values outside the diagonal are very low and balanced.

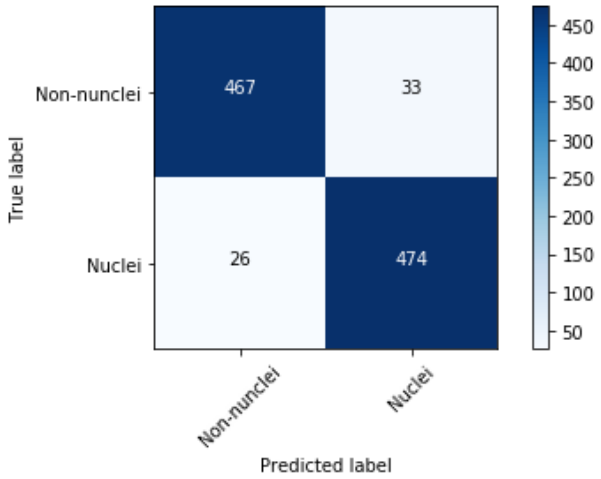

Figure 6. SGAN Confusion Matrix without normalization

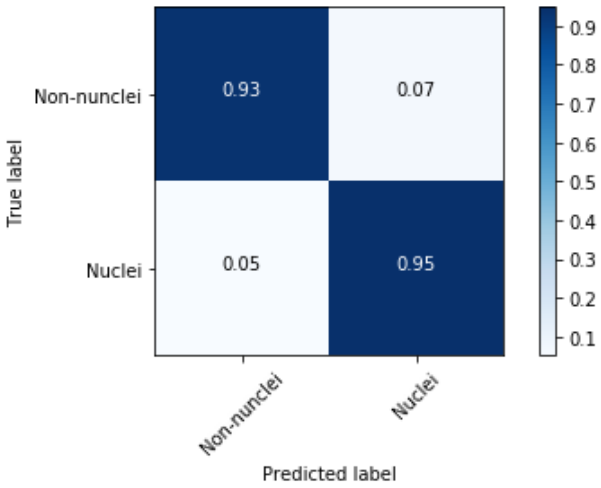

Figure 7. SGAN Normalized Confusion Matrix

We computed the precision, recall, F-measure and support for each class (see Table 2). A precision of 94% is intuitively a good ability of D to not to label as nucleus a sample that is not a nucleus. A recall of 94% is intuitively a good ability of D to find all the nucleus samples.

The F-measure can be interpreted as a weighted harmonic mean of the precision and recall, where an F-measure score reaches its best value at 1 and worst score at 0. In this case, the F-measure was not weighed *w.r.t.* any factor, i.e. $\beta=1$. Table 2 shows the full classification report for our SGAN's D model.

Recall that the SSAE+SMC model used a different approach and dataset to calculate its Precision, Recall, F-measure and Average Precision (*AveP*) [1]. However, the classification task was somehow similar, since both models (SSAE+SMC and SGAN) output the probability of the input image patch corresponding to a nucleus or not through a Softmax Classifier. That is why we considered pertinent to compare its values (see Table 3).

Table 1. SGAN's D Classification Report

	Precision (%)	**Recall (%)**	**F-measure (%)**	**Support (%)**
Non-Nucleus	95	93	94	500
Nucleus	93	95	94	500
Avg / Total	94	94	94	1000

Table 2. The mean associated with Precision, Recall, F-measure and Average Precision (AveP) for SSAE+SMC, and SGAN's D

Model	**Precision (%)**	**Recall (%)**	**F-measure (%)**	**AveP (%)**
SSAE-SMC	88.84	82.85	84.49	78.83
SGAN	**94**	**94**	**94**	**93.2**

VI. Conclusion and Future Work

In this paper, a Semi Supervised Generative Adversarial Network (SGAN) is presented for automated nuclei detection on breast cancer histopathology. We have shown how the model can capture high-level feature representations of pixel intensity in a semi-supervised manner. 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. To show the effectiveness of the proposed framework, we compared it with the SSAE model. Regardless of the difference in the testing approach and dataset that both models used, the SGAN model appear to suggest that it works well in learning useful high-level features for better representation of nuclear structures. However, we consider that these evaluations are not sufficient to determine that it does or can outperform state-of-the-art methods on nuclei detection.

We are eager to explore the following related ideas to improve our results and arrive to more certain conclusions:

- Replicate the testing procedure from Xu et al. [1] and implement the same qualitative, quantitative and sensitivity analysis to have a better and deeper perception when comparing both models.
- Improve SGAN model training by evaluating the impact of recently published techniques such as: One side label smoothing, Historical averaging, Virtual Batch Normalization (VBN), and Inception Scoring proposed by Salimans et al. [3], feature matching, and adding artificial noise to inputs [18][19][20].

Acknowledgment

We thank Dr. Juan Shan, Assistant Professor at Pace University for sharing her pearls of wisdom and expertise during the course of this research.

References

[1] 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.

[2] A. Radford, L. Metz, & S. Chintala. (2016, Jan.). "Unsupervised representation learning with deep convolutional generative adversarial networks." ArXiv E-Prints. [On-line]. Available: http://arxiv.org/ abs/1511.06434 [1 May 2018].

[3] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, & X. Chen. (2016, Jun.). "Improved techniques for training GANs." ArXiv E-Prints [On-line]. Available: https://arxiv.org/abs/1606.03498 [7 May 2018].

[4] 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].

[5] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath, "Generative Adversarial Networks: An Overview," in *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53-65, Jan. 2018.

[6] E. Denton, S. Chintala, A. Szlam, & R. Fergus. (2015, Jun.). "Deep generative image models using a Laplacian pyramid .of adversarial networks." ArXiv E-Prints. [On-line]. Available: http://arxiv.org/abs/1506.05751 [7 May 2018].

[7] I. J. Goodfellow. (2015, May). "On distinguishability criteria for estimating generative models." ArXiv E-Prints. [On-line]. Available: https://arxiv.org/abs/1412.6515 [7 May 2018].

[8] D. J. Im, C. D. Kim, H. Jiang, & R. Memisevic. (2016, Dec.). "Generating images with recurrent adversarial networks."ArXiv E-Prints. [On-line]. Available: https://arxiv.org/abs/1602.05110 [8 May 2018].

[9] D. Yoo, N. Kim, S. Park, A.S. Paek, & I.S. Kweon. (2016, Nov.). "Pixel-level domain transfer." ArXiv E-Prints. [On-line]. Available: https://arxiv.org/abs/1603.07442v2 [7 May 2018].

[10] S. Ioffe & C. Szegedy. (2015, Mar.). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." ArXiv E-Prints [On-line]. Available: https://arxiv.org/abs/1502.03167 [7 May 2018].

[11] D. P. Kingma & M. Welling. (2014, May). "Auto-encoding variational Bayes". ArXiv E-Prints [On-line]. Available: https://arxiv.org/abs/1312.6114 [7 May 2018].

[12] D. P. Kingma, D.J. Rezende, S. Mohamed, & M. Welling. (2014, Oct.). "Semi-supervised learning with deep generative models." ArXiv E-Prints [On-line]. Available: https://arxiv.org/abs/1406.5298 [8 May 2018].

[13] D. Rezende, S. Mohamed, & D. Wierstra. (2014, May). "Stochastic backpropagation and approximate inference in deep generative models." ArXiv E-Prints [On-line]. Available: https://arxiv.org/abs/1401.4082 [7 May 2018].

[14] A. Odena (2016, Oct.). "Semi-supervised learning with generative adversarial networks." ArXiv E-Prints [On-line]. Available: https://arxiv.org/abs/1606.01583 [7 May 2018].

[15] V. Vargas & J. Koller. "GAN-for-Nuclei-Detection." Internet: https://github.com/vmvargas/GAN-for-Nuclei-Detection, May 7, 2018 [7 May 2018].

[16] E. Linder-Norén. "Keras-GAN." Internet: https://github.com/eriklindernoren/Keras-GAN/tree/master/sgan, April 20, 2018 [7 May 2018].

[17] D.P. Kingma, & J. Ba. (2017, Jan.). "Adam: A method for stochastic optimization." ArXiv E-Prints [On-line]. Available: https://arxiv.org/abs/1412.6980 [2 May 2018].

[18] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, & F. Huszár. "Amortised MAP inference for image super-resolution." ArXiv E-Prints [On-line]. Available: https://arxiv.org/pdf/1610.04490v1.pdf [2 May 2018].

[19] Y. Mroueh, T. Sercu, & V. Goel. (2017, Jun.). "MCGAN: Mean and covariance feature matching GAN." ArXiv E-Prints [On-line]. Available: https://arxiv.org/abs/1702.08398 [7 May 2018].

[20] M. Arjovsky & L. Bottou. (2017, Jan.). "Towards principled methods for training generative adversarial networks." ArXiv E-Prints [On-line]. Available: https://arxiv.org/abs/1701.04862 [2 May 2018].