

# Wasserstein Deep Convolutional Generative Learning Adversarial Network for Dynamic Sparse Sampling (WDC-GLANDSS)

David S. Helminiak,<sup>\*</sup> Richard J. Povinelli,<sup>\*</sup> and Dong Hye Ye<sup>\*</sup>

*Opus College of Engineering, Marquette University, Milwaukee*

E-mail: david.helminiak@marquette.edu; richard.povinelli@marquette.edu;  
donghye.ye@marquette.edu

## Abstract

Within the machine learning community, Deep Convolutional Generative Adversarial Networks (DCGANs) and Wasserstein Generative Adversarial Networks (WGANs) have become increasingly researched and implemented frameworks for unsupervised image learning tasks. The application of the underlying component Neural Networks (NNs), the generator and discriminator, can better inform more traditional supervised learning tasks, particularly cases requiring label prediction. This paper combines these frameworks together as to gain insight as to the robustness of a generalized architecture for sparse sampling of medical and material samples: a Wasserstein Deep Convolutional Generative Learning Adversarial Network for Dynamic Sparse Sampling

---

## ORCID

David S. Helminiak: 0000-0001-7637-471X

## Notes

The authors declare no competing financial interest.

(WDC-GLANDSS), and compares the results against an existing linear approach to the same applications: a Supervised Learning Approach to Dynamic Sampling (SLADS). This comparison shows SLADS to be highly successful at performing sparse sampling and the possible benefits that the WDC-GLANDSS can offer with further development and optimization.

## Introduction

Supervised Learning Approach to Dynamic Sampling (SLADS) was originally developed in order to reduce image acquisition time for applications requiring point-wise scanning. This would result in increased throughput for scanning equipment, while also potentially reducing exposure of critical samples to radiation (depending on the method being employed). Traditional point-wise image scanning is typically conducted along a rectilinear grid; i.e. raster scanning, with the path statically determined before sample acquisition. Depending on the scanning equipment and methodology (typically forms of microscopy and spectroscopy) and desired final image fidelity levels, single measurements can require anywhere from 0.5 to  $\sim 5$  seconds.

Sparse sampling, as used in SLADS, collects only a subset of pixels, from which the full image may be reconstructed through simple gradient-based interpolation. Further, the technique is dynamic, attempting to make optimal choices for scanning locations, as informed by previously determined values. As a result, SLADS can be seen to essentially perform a real-time version of the functionality used by traditional thresholding algorithms, albeit with an additional structurally aware component. Besides its originally intended functionality, this makes it uniquely suited for any scanning application where known information of interest needs to be extracted from background noise.

One possible situation this may be applicable to is in material characterization. The Marquette University Shock Physics Lab currently performs uniaxial strain and oblique impact studies to characterize stress and strain states for various materials. This requires

the measurement of transverse and shear velocities during high speed impacts as produced using a pneumatic accelerator. These experiments take significant amounts of time in both execution and preparation of material components. This means that simulations are an extremely useful step in ensuring the maximization of information gain from each intended experiment.

These peridynamic simulations rely on precise definitions of the intended materials, which for complex structures, specifically the grain sizes and orientations of sandstone samples (currently utilized by the lab), presents particular difficulty. The high variation in the underlying, non-homogenous chemical makeup means generalization should only be performed with a sufficient number of samples characterized in an identical manner. While high resolution electron microscope imaging can capture the information, the specific extraction has not been able to be reliably performed through typical thresholding algorithms, due to the significant intra-class variation. While the grain structures themselves are distinguishable according to their boundaries, they can contain component pixel values of the same level as background noise. Manual determination of grain characteristics can be inconsistent and prohibitively expensive in terms of time. These restrictions can be potentially overcome though the use of structurally-aware algorithms such as SLADS to isolate pixels that contain relevant information.

SLADS functionality does have a limitation imposed by its reliance on a linear model to describe image features. Continual addition of polynomial features, to the linear regression, causes rapid increases in model specificity and overall decreased capability for generalization. Similarly, the model's complexity increases linearly with the number of training samples provided, which limits its application to smaller and clean training sets. For a noisy, but large dataset, a more complex modeling architecture, as presented by advanced Neural Network (NN) machine learning algorithms, specifically the Generative Adversarial Network (GAN) architecture proposed by Ian Goodfellow et al.<sup>1</sup> in 2014, could offer significant advantages. The GAN combines a primary generative NN that forms images from a random input

vector, with a secondary discriminator NN that provides feedback as to the relative success of the produced output. This work advances the characterization and understanding of the limitations presented by the SLADS model and proposes a new variation: Wasserstein Deep Convolutional Generative Learning Adversarial Network for Dynamic Sparse Sampling (WDC-GLANDSS) based on more advanced forms of the GAN architecture: Wasserstein GAN (WGAN) and Deep Convolutional GAN (DCGAN), to improve the reconstruction modeling phase of the SLADS formula.

In this work, SLADS has been shown to be highly successful in performing sparse sample reconstruction of both biological tissues and non-homogeneous material. While the architecture for WDC-GLANDSS has been well developed herein, it remains highly sensitive to hyperparameter variation and the applied datasets with negligible results for both intended sets, though demonstrated success with a more basic subset of the MNIST database.

## Prior Work

The formalized SLADS framework was introduced in 2018 by Godaliyadda et al.,<sup>2</sup> though a similar approach had been applied in 2016 for X-ray crystalline protein acquisition.<sup>3</sup> Godaliyadda et al. focused on the reduction of measurement times associated with imaging processes and related radiation exposure levels. As a dynamic approach SLADS ideally selects either singular, or groups of points most likely to reduce the total distortion in a reconstruction of a full scan. The technique was shown through simulations, for Scanning Electron Microscope (SEM) samples, to significantly lower error in produced reconstructions, compared to both random and low-discrepancy sampling techniques. The methodology was adopted for Raman spectroscopy<sup>4</sup> later that year, yielding 7x improvement in the number of pixel measurements needed to give an  $\sim 0.1\%$  difference in error relative to full scans. Both of these prior works make note that SLADS' calculation times of  $\sim 100$  ms, for determining successive pixels to scan, are negligible when compared against its total acquisition

time reduction. Additional research is currently being conducted to apply this technique to Nanospray-Desorption Electrospray Ionization (nano-DESI) procedures to enable higher experimental throughput. However, this work recognizes that the SLADS linear regression model may not function sufficiently for such an application, given the higher number of features in the intended Mass Spectroscopy Imaging (MSI) procedures. This suggests a possible need for employment of more advanced algorithms.

GANs have become an increasingly researched and implemented framework for unsupervised learning tasks since their 2014 introduction<sup>1</sup> and combine a pair of NNs together. The generator NN accepts a vector of noise (usually Gaussian, or uniform) as an input and performs weighted transforms to match it with trained features. The discriminator NN then performs a classification to determine if the generated image matches with the expected feature set of authentic images, providing remedial feedback to the generator.

These have only recently had noteworthy success for 2-dimensional image inputs through augmentation with Convolutional Neural Networks (CNNs). This combination framework, DCGAN, was proposed by Radford et al.<sup>5</sup> in 2015 and has been shown to be highly effective in producing photorealistic images. It further has the benefit of being able to specify semantic qualities, in the generated output, which can be greatly helpful for dataset augmentation. The framework introduced a set of optimized hyperparameters and used convolutional windows to find areas of spatial correlation, which in turn allow for greater variation in 2-dimensional data, particularly images and videos.

Generally, GANs are noted as being relatively unstable during training because of the alternating losses between the generator and discriminator networks. This also makes it more difficult to discern whether the networks have actually converged. Partly this is due to the discriminator becoming effective too quickly as compared to the generator. With minimal levels of feedback passed to the generator, it cannot effectively converge. A second reason for the instability can be seen for the use of the GAN framework's originally proposed Jensen-Shannon divergence loss function employed to determine the difference between real

and synthetic images. These issues and possible solutions, combined in the Wassertein Generative Adversarial Network (WGAN) framework, proposed in 2017 by Arjovsky et al.<sup>6</sup> and noted by Wolterink et al.<sup>7</sup> in 2018 for application on biomedical images specifically.

The former effect can be limited by replacing the generator loss function to minimize the initial ability for the discriminator to tell the difference between synthetic and real images. The second can be mitigated through the replacement of the calculation for image differences (loss functions) with the Wassertein, or earth mover's distance metric, that determines the amount of work needed to transform one image to another, made computational through the Kantorovich-Rubinstein duality principle. This necessitates additional clipping of the discriminator weights in order to comply with 1-Lipschitz function constraints. While this does improve network functionality, changes to the clipping boundaries have been noted to cause large variations in network performance. However, adding a gradient penalty to the clipping levels, as proposed by Gulrajani et al. in 2017<sup>8</sup> removes the issue of needing to tune the hyperparameter entirely.

Overall, these changes have the additional effect of helping prevent to mode collapse (high reduction in variation of the generator's produced fake images). This could also potentially be further improved by incorporating batch variation levels into the discriminator's loss function.

The majority of medical research into the application of GANs involves the production of low-error reconstructions to minimize radiation exposure in samples for Computed Tomography (CT) scans. A 2017 work by Wolterink et al.<sup>9</sup> uses low radiation CT scans with a generator CNN to upsample captured images back to standard dosage levels. An alternative method proposed by Nie et al.,<sup>10</sup> also in 2017, uses Fully Convolutional Networks (FCNs) to generate pseudo CT scans using Magnetic Resonance Imaging (MRI) results passed through the GAN framework to achieve more realistic results. The same year, Schlegel et al.<sup>11</sup> used a GAN to annotate images of optical tissues with a score denoting the presence of notable structural anomalies. These techniques are also slowly being applied into other scanning

technologies as in the case of Kim et al.’s<sup>12</sup> 2018 work to use a set of varying contrast low-resolution MRI scans to inform singular higher-resolution reconstructions.

Regarding the secondary intended application of this work, there exists an exhaustive list of automatic image thresholding techniques and variations, as would be needed for the intended segmentation of grain boundaries. These are split according to the survey conducted by Sezgin et al.<sup>13</sup> in 2004 into basic categories including: local, spatial, object attribute, clustering, histogram, and entropy. While the study identified specific entropy and clustering algorithms that performed best for grayscale text, the study did not extend to images with significant intraclass variation. Putting aside pixel intensities and gradients, noting there exists an uneven spread of light across the samples intended for segmentation, this can limit algorithms for investigation to those with some level of structural awareness. The most prominent of these noted in the 2004 study rely upon distance metrics in combination with general image statistics. However, as grain boundary positions are by their nature not regularly interspersed, existing automatic methods can be seen to be ineffective, although possible with manual optimization.

## Methodology

### Datasets

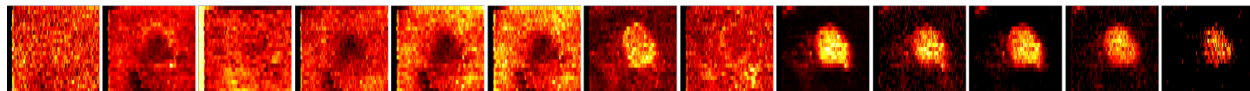


Figure 1: Every other discretization of a single pancreatic islet sample.

For examining the effectiveness of SLADS and WDC-GLANDSS at improving scan times of medical samples, a collection of 8 pancreatic islets each with 21 lines/rows of continuous data was used. The original scans have a mass range of 100 to 2000 m/z and were collected at sampling rates conducted between 5 and 20  $\mu\text{m/s}$  with an equipment acquisition rate of 130 ms per spectra. These continuous sets were each discretized to 26 images using

MSI Quickview software, providing 208 training examples (Figure 1). All of the actual tissue samples are positioned roughly in the center of each image giving a high degree of positional predictability. However, approximately half of the discretizations do not show visual distinction against the background.

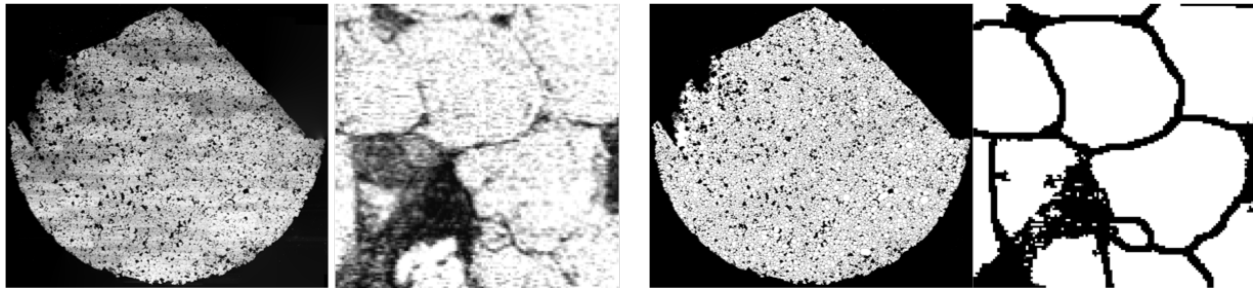


Figure 2: A full and close-up of sandstone sample scanned through an electron microscope on the left (a) and on the right (b) a manually masked full and close-up of the same sandstone sample.

Comparisons for the algorithms’ effectiveness, for thresholding materials, were conducted on sandstone samples obtained from Mosinee, Wisconsin’s Copper Mountain location. These were pre-processed to have with thicknesses from 0.6 to 2.5 mm with the understanding that they are intended for experimental impact velocities of 50 - 200 m/s. Simulations are intended to be conducted using Peridigm and CTH software and are intended to extract sandstone strength characteristics to inform the simulated material’s behavior in planar and oblique impact experiments. A single high-resolution (2  $\mu\text{m}/\text{pixel}$ ) 10485x9014 pixel resolution ( $\sim 21 \times 18$  mm) image, was obtained with a Jeol JSM-6510 SEM, along with a manually masked version (Figure 2). These were cropped to a symmetric 8192x7168 pixels, to eliminate image borders containing no meaningful information, yet allowing for symmetric 128x128 pixel segmentations. If the average pixel intensity, in the manually masked segmented image, was less than, or equal to that of the binary version of the original, then they were both judged to contain little to no meaningful information and were therefore discarded from the formed dataset. This procedure yielded a dataset of 2473 images of which 208 were randomly selected to provide reasonable compute times and a more accurate comparison of performance as compared to the pancreatic set. These images rarely include any consistency in terms of a



positional reference and generally poor contrast against the background. Further, the complete SEM acquisition was performed with noticeable vibrations being caused from nearby construction, which led to certain areas being particularly blurry. A general summary of the specific dataset parameters used may be found in Table 1.

Table 1: Dataset information and final hyperparameters after WGAN framework integration.

Dataset	Image Size	Number of Samples			Learning Rate	Weight Clipping
		Total	Train	Test		
MNIST	28x28	600	408	NA	0.001	0.001
Pancreatic	28x28 (SLADS)	208	166	42	0.001	0.001
	64x64 (WDC-GLANDSS)					
Sandstone	128x128	208	166	42	0.0008	0.00001

## WDC-GLANDSS

## Architecture

### Approach

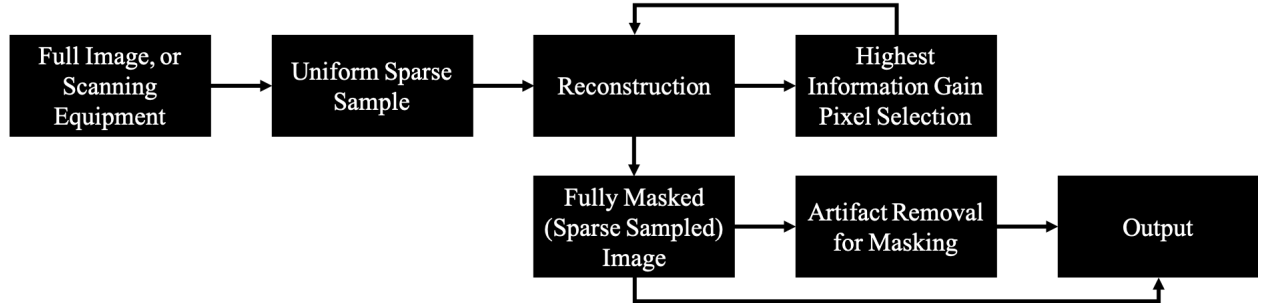


Figure 3: Generalized approach for conducting sparse sampling and thresholding operations.

Both SLADS and WDC-GLANDSS make use the same overall procedure, visualized in Figure 3, to conduct sparse sampling and thresholding applications. Taking a uniform sparse scan from the full sample, the trained models are used to perform a reconstruction. Pixels that provided the highest level of Information Gain (IG) are selected until termination criteria have been satisfied. For SLADS, the IG comprises the Estimated Reduction in Distortion (ERD) levels present in the reconstruction, while for WDC-GLANDSS it is solely based on

maximal pixel intensity levels. This procedure ensures that with the appropriate termination conditions and optimized loss functions that only locations with meaningful information, that is to say non-background values, will be acquired. The relative effectiveness of each program relies on their respective ability to perform the image reconstruction with a limited set of information.

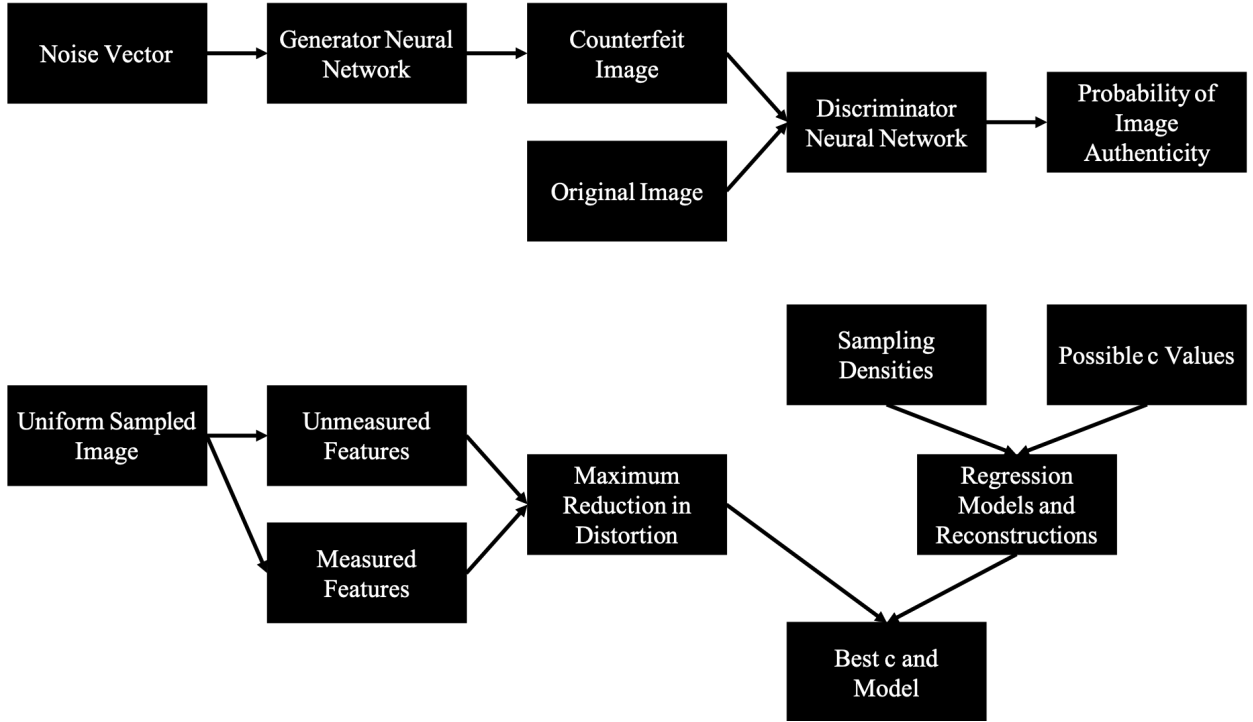


Figure 4: Architecture of sampling training procedures for top: (a) SLADS and bottom: (b) WDC-GLANDSS.

## SLADS

SLADS employs a linear regression model (from the sklearn v0.20.3 Python package), consisting of a polynomial features list "theta," to reconstruct a full sample. After taking the initial uniform sample, subsequent pixels are scanned in either singularly, or according to a predetermined batch size, in order to maximize the reconstructed sample's ERD; i.e. the difference in distortion present in the existing reconstruction, and that present in a reconstruction made with a considered pixel. This distortion level prediction is aided through

the application of Gaussian kernels to scanned pixels. The size of these applied kernels is optimized according to the distance between an un-scanned pixel and the nearest scanned neighbor, limited by a controlling constant "c." This value may be optimized during training, but remains constrained to a set provided by the user as a more complete search would be computationally prohibitive. Although CPU multi-threading was added to great relative effect for training times, optimization of a model using 200+ 128x128 pixel images can take upwards of 24 hours on a 32-thread 3.0 GHz dual-socket workstation for 7 initial sampling densities and 5 possible "c" values.

For each proposed starting sampling density (user specified vector), a uniformly sampled random mask for each of the training images is created. This base image, containing actual values specified by the mask, is to be used for the simulation of adding polynomial features to the linear model and effect of varying the "c" value. For each of the images and for each of the masks previously generated, the polynomial features of the initially unmeasured points are loaded and added to a new regression model. These models are then saved and evaluated over each of the training images from which a best "c" value and its corresponding "theta" set are determined for use in testing, or experimental application. The overall architecture may be seen in Figure 4 (a), with more formal definition of the specifically employed equations provided by Godaliyadda.<sup>2</sup>

## WDC-GLANDSS

WDC-GLANDSS (implemented with TensorFlow v1.13 in Python with eager execution) comprises a combination of the DCGAN and WGAN architectures in order to produce a generator network capable of producing realistic samples from normally distributed noise, as well as a discriminator that determines relative loss values for the difference between generated and original images. For actual reconstruction, the generator itself is augmented by a secondary gradient descent procedure that optimizes an initially normal distribution random input vector, such that the produced reconstruction more closely adheres to already

scanned values. This later procedure modifies a linear model containing weight and bias values for each pixel using the discriminator to determine loss levels, an Adam optimizer, learning rate of 0.01, and a maximum run time of 200 epochs.

The underlying DCGAN architecture of WDC-GLANDSS, seen in Figure 4 (b), follows the guidelines proposed by Radford et al.<sup>5</sup> with strided convolutions instead of pooling functions, fully connected input and output layers, batch normalization (at all except the discriminator input and generator output layers), ReLU activations for all the generator layers (using tanh at the output), and LeakyReLU for all layers in the discriminator. Initially used were the suggested LeakyReLU dropout value of 0.2, learning rate of 0.0002, and adjusted beta of 0.5 in combination with an Adam optimizer in place of stochastic gradient descent. Convolutions were consistently performed with a kernel window size of 5x5 pixels and a typical stride value of 2 pixels. The generator loss was calculated, as the sigmoid cross entropy, between the generated image and an array of ones. Using the same function, the discriminator adds the determined real losses (between an array of ones and the original images) and the generated losses (between an array of zeros and the generated images).

Due to the observation of modal collapse and high variance in generator/discriminator losses during initial testing, modifications were eventually made to these values to come into accordance with the WGAN approach.<sup>6</sup> The Adam optimizer was replaced by a Root Mean Squared (RMS) optimizer that was not governed according to a momentum factor. The more optimal gradient penalty modification<sup>8</sup> was not implemented due to project time constraints, with the discriminator weights clipped manually during each epoch. These hyperparameters still remained quite difficult to scale appropriately between the applied datasets. In the case of the pancreatic set the best performance obtained was at a reduced 28x28 pixel resolution, though the full 64x64 pixel images were able to run SLADS. Due to the lower number of prominently identifiable structures, the Sandstone set was unable to be successfully optimized to any recognizable degree and so was left at its 128x128 pixel resolution for both models. The batch size remained constant at 100 given the similar numbers of training samples.

Variations of these parameters for each dataset are specified in Table 1. The loss functions were also altered, the generator’s becoming the negative mean of the produced image, while the discriminator’s changed to be the difference between the means of the generated and original images.

GPU acceleration with TensorFlow’s relatively recent integration of eager execution functionality was employed to aid computation times. Though this does limit the overall model size given limited on board memory, 500 epochs were able to be completed for 200+ 128x128 pixel training images in about 4 hours on a single Nvidia GTX 1080 Ti. In order to establish an upper bound on the model size, the training/testing images’ original resolution is never exceeded by the model, but automatically decreased should the system requirements make it necessary. Starting with the image set’s row and column dimensions, the larger of the two is set as the maximum desired symmetric dimension ( $dim_{XY}$ ). The initial starting size for the generator is minimized (Min) using Equation 1 (using rounding to the nearest decimal place (RD) to mitigate rounding errors), with a temporary start size (TSS) ranging from 1 to whole values less than the original symmetric dimension. The selection ensures that the minimum number found, in the array, contains a whole value that can be multiplied by a factor of 2 back up to the maximum allowable symmetric dimension. The number of layers in the generator and discriminator networks, then becomes equivalent to the number of times this starting size must be doubled to reach this maximal value. Should the maximal size not be a whole number at any layer, it is decreased to the nearest even value until it does so.

$$Min([RD((\log_2(dim_{X_Y}) - \log_2(TSS))) - int(RD(\log_2(dim_{X_Y}) - \log_2(TSS)))) == 0]) \quad (1)$$

Should the memory space required to hold a model with these values exceed that of the system, the dimensions are continually decreased to the next even value, recalculating the starting generator dimensions and layer count. As TensorFlow’s eager execution model calculates values immediately rather than forming graphs, the limitation for a system can

be found as the largest layer within any given NN model. For the architecture specified this can be approximated by the area of the maximal symmetric dimensions, times that occupied by the generator's starting dimensions, times the maximal total pixel depth, times the space occupied by each node. It was found that for each tensor node needed in the network could be approximated to require 32 bytes. The maximal total pixel depth is determined statically through the original pixel resolution (256) times the number of channels (3 for color, 1 for grayscale).

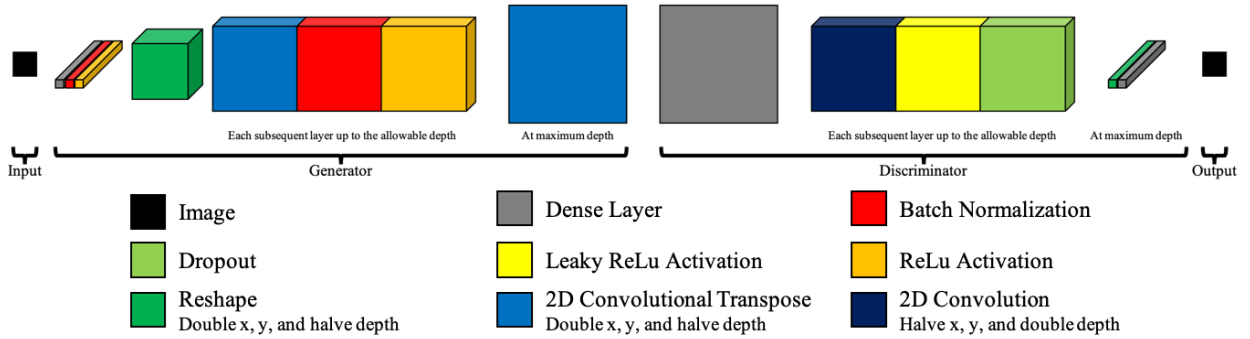


Figure 5: Final combination of generator and discriminator networks for WDC-GLANDSS architecture

The final arrangement for the WDC-GLANDSS architecture can be seen in Figure 5. The generator reshapes the image dimensional input to a singular vector of length equal to the starting area (square of the symmetric starting size) times the maximal total pixel depth. Performing batch normalization and the ReLU activation function, the vector then is reshaped into an array with symmetric dimensions (given by the determined starting size) and a pixel depth equal to the maximal total pixel depth. For each of the layers, this depth is divided by 2, with a convolutional transpose window being applied. The final layer again performs the convolutional transpose, but with a stride value of only 1. The generator therefore produces an image with the maximal allowable symmetric dimensions and a depth equal to the number of channels in the original training images. The discriminator network accepts any image with this same set of dimensions, but performs the reverse operations at each layer. An additional convolutional layer is added immediately after the input with a

stride of 1 to produce a symmetric NN model. The penultimate layer in the discriminator therefore has the symmetric generator starting size dimensions with a depth of the maximal total pixel depth. The array then is flattened and passed to a dense layer with a single node for evaluation.

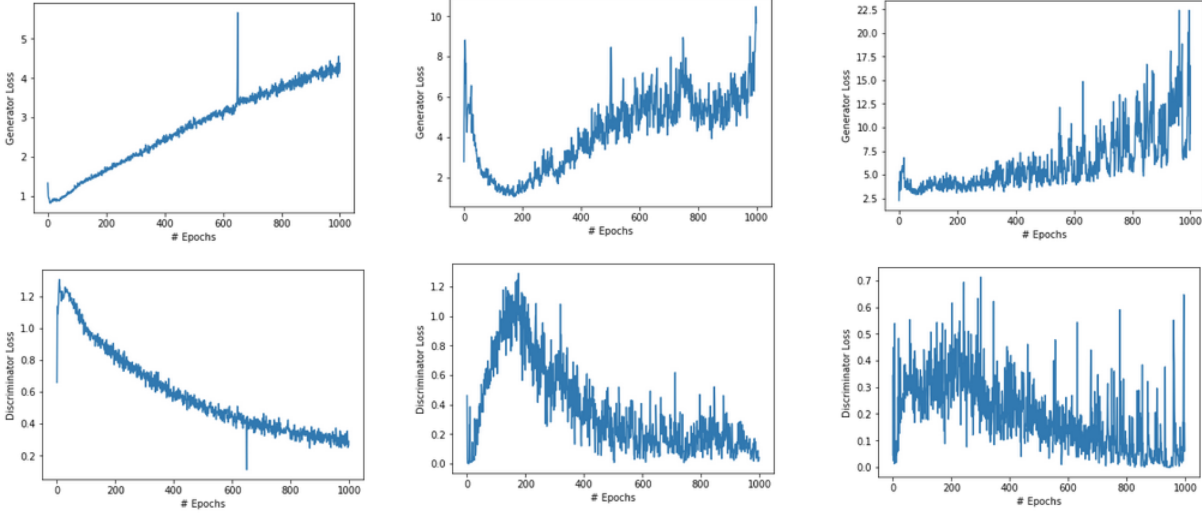


Figure 6: Progression of underlying DCGAN architecture development with the use of the MNIST dataset with respect to the generator and discriminator losses. On the left (a) can be seen the number of layers in discriminator and generator limited by input resolution; in the center (b) an augmented layer count to produce linear progression of pixel depth in both generator and discriminator and on the right (c) the discriminator layer count limited by input resolution with generator layers augmented to produce linear progression of pixel depth.

A subsection of the MNIST (Mixed National Institute of Standards and Technology) numerical handwriting dataset<sup>14</sup> containing 600 28x28 images was used for initial testing and optimization of the DCW-GLANDSS architecture, prior to the alterations made to incorporate WGAN approaches. This set was used in order to minimize the required time between runs given the higher image resolutions in the pancreatic and sandstone datasets. Training time using an Nvidia 1080 TI 11 GB GPU and an image batch size of 10, took  $\sim 80$  minutes to complete for each of the following simulations. Figure (a), shows an example of the discriminator and generator losses relative to the number of training epochs. The generator loss continues to increase as it becomes increasingly difficult to deceive the trained discriminator

network. Theoretically, when the discriminator is trained to its maximal degree, as close to a loss of 0 as possible, the generator loss will begin to decrease to its minimal value. However, the training time needed to do so is prohibitive to most applications, such that a cap on the number of epochs, or another termination criteria remains a necessity.

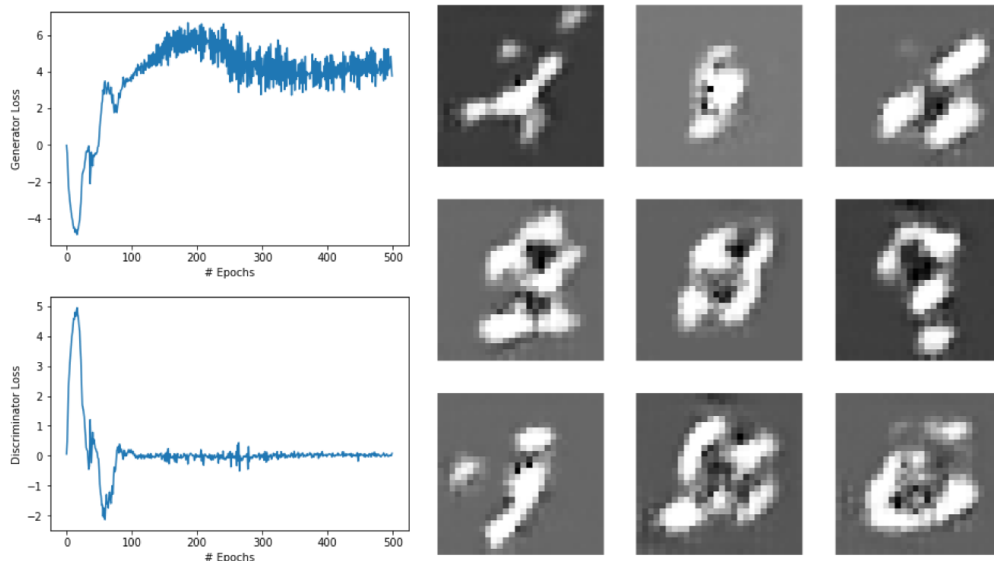


Figure 7: Resulting generator and discriminator losses for MNIST dataset after WGAN modifications on the left (a) and a resulting set of images produced by the generator with random noise input after 500 epochs on the right (b).

Adding additional layers with a stride value of 1 and continually decreasing depth to the generator as well as increasing depth to the discriminator in order to give a smoother transition between pixel depths results in an initially opposite relationship. Seen in Figure 6 (b), the generator loss continually decreases, until the discriminator's begins to minimize. The variation in discriminator loss also becomes more significant and subjectively detrimental to the training process. In instances where the variations settle, mode collapse was the common result. Attempting to obtain a more joint convergence for the generator and discriminator, with respect to the training time, the additional layers were left in the generator and removed from the discriminator, resulting in Figure 6 (c). However, the model produced minimal convergence in both the discriminator and generator, having even more variation in losses. This suggests there is no clear value in having smooth transitions of pixel depth between



layers in either underlying NN given the automated generation of parameters previously described. After the WGAN approaches had been incorporated into the architecture and increasing the batch size to 100 samples, the results in Figure 7 were obtained with massive improvements to the model convergence and produced images.

## Results

### Evaluation

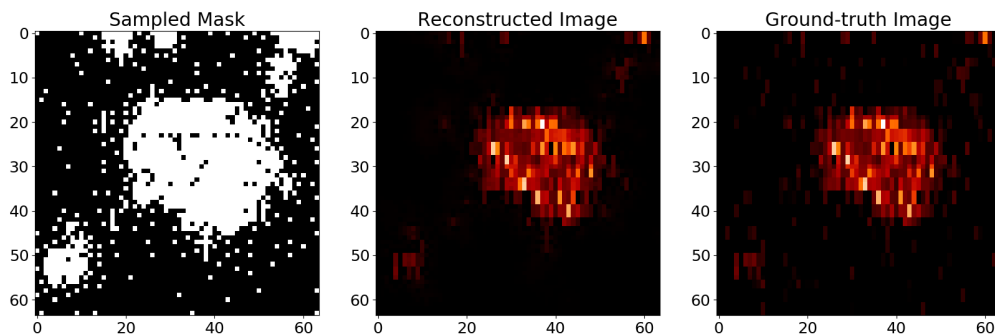


Figure 8: Sample reconstruction of a discretized pancreatic islet using SLADS with 30% of pixels sampled resulting in an MSE of 35.66 and SSIM of 0.44.

Establishment of the reconstruction and masking capabilities for SLADS and WDC-GLANDSS was performed with Mean Square Error (MSE) (Equation 2) and Structural Similarity Index (SSIM) (Equation 3). The datasets were randomly split into 80% training and 20% testing sets, then being used to train SLADS and WDC-GLANDSS models. A sample reconstruction of a pancreatic islet performed using SLADS can be seen in Figure 8. WDC-GLANDSS model convergence was visualized through generator/discriminator loss plots. An equivalent model of convergence for SLADS was obtained through performing 50% sparse sampling reconstructions across the entire testing set after the addition of each training sample’s features to the linear model. The reconstructions’ MSE/SSIM levels were

then averaged together and plotted.

$$MSE = \frac{1}{n} \sum (y - \hat{y})^2 \quad (2)$$

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2\mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3)$$

### **Pancreatic Islets Dataset**

Seen in Figure 9 (a), SLADS converged very quickly for the biological tissue, with more than about 25 training samples only causing variations in the MSE/SSIM levels. The convergence remains inferior to peak values obtained with only a couple of samples. Nonetheless, Figure 9 (b) illustrates that the program is capable of producing high SSIM, low MSE full reconstructions of pancreatic samples with up to a 70% reduction in the number of pixels needed for acquisition. In contrast, while the equivalent WDC-GLANDSS model (Figure 10) was able to converge both its generator and discriminator networks to a reasonable degree, the MSE/SSIM reconstruction results were very poor. No appreciable improvement can be seen with respect to the number of pixels sampled within the images, nor are there any noteworthy variations in the obtained information.

### **Sandstone Dataset**

The reconstruction results for the sandstone dataset are nearly identical to those determined for the pancreatic set. SLADS training, in Figure 11, ceased improvement after roughly 50 training images with peaks above its convergence level. Yet the model still produced highly accurate reconstructions with a sparse set of pixels. The progression of the model's decreasing MSE and increasing SSIM for each sampling level can be observed to generally fit a normal distribution making the number of pixels required for specific fidelity targets theoretically quite predictable. WDC-GLANDSS results in Figure 12 also fail to show meaningful progression, though there are a larger number of outlying points in the MSE/SSIM

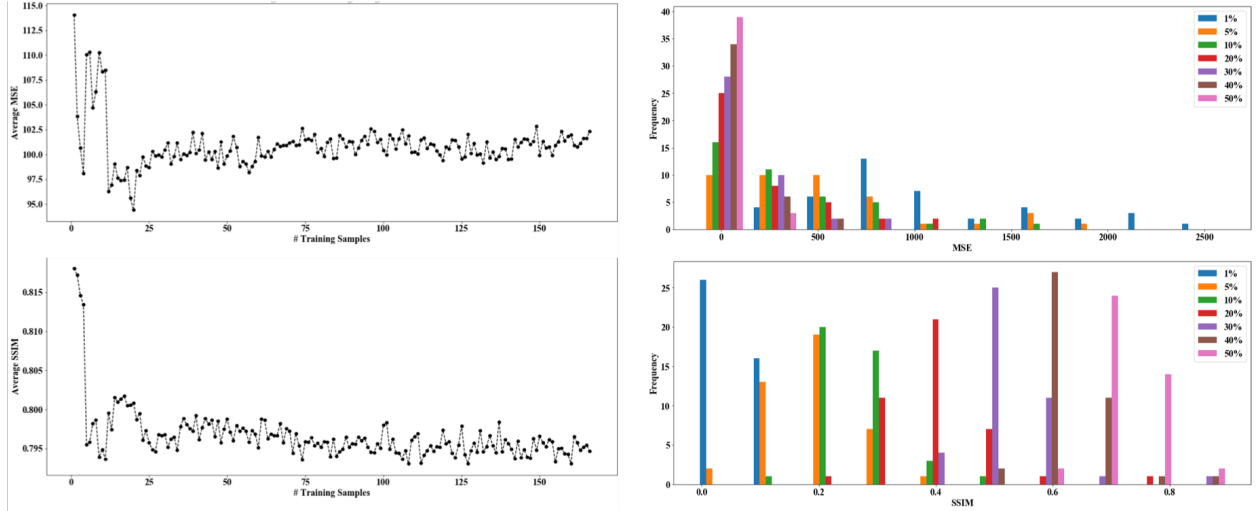


Figure 9: SLADS performed on the pancreatic islets dataset showing on the left (a) MSE and SSIM convergence and on the right (b) the distribution of MSE and SSIM results relative to the sampling percentages used for reconstruction.

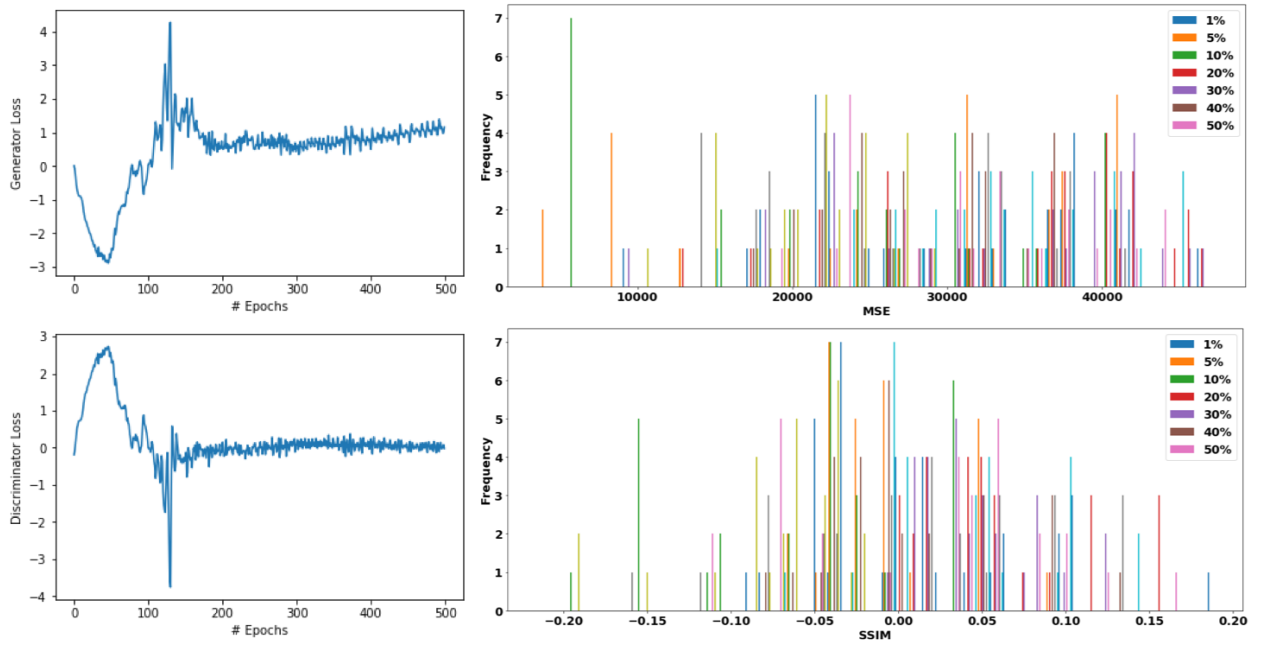


Figure 10: WDC-GLANDDS performance on the pancreatic islets dataset with generator/discriminator losses on the left (a) and the resulting MSE/SSIM levels on the right (b).

distributions then seen for the pancreatic samples.

These behaviors continued for the theorized thresholding application visualized in Figure 13. Looking at the MSE/SSIM between the generated binary masks and those manually formed, SLADS can be characterized as highly applicable and capable, but WDC-GLANDSS

as wholly ineffective.

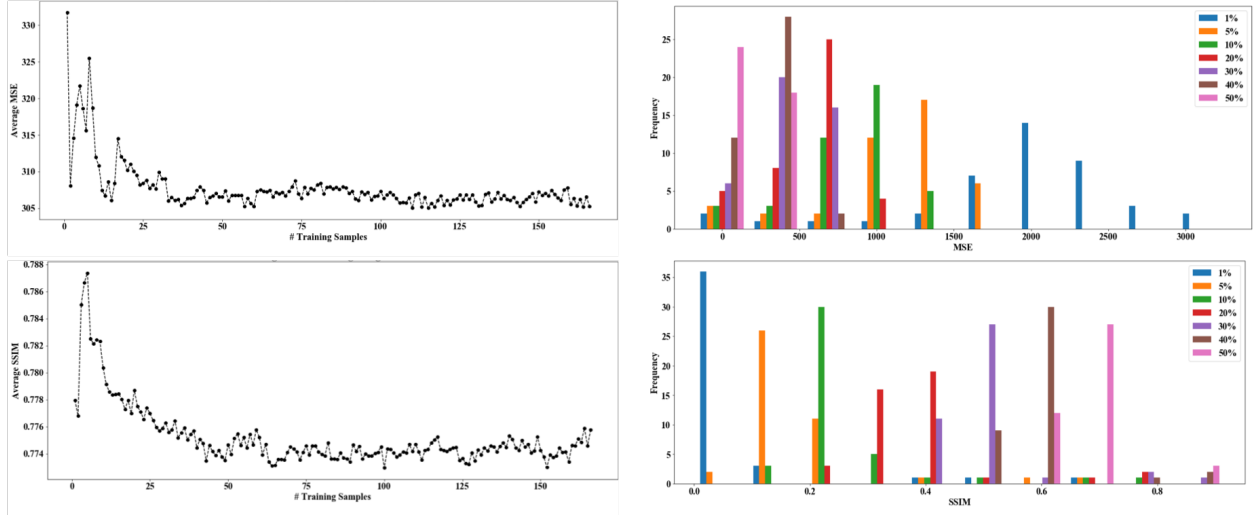


Figure 11: Sandstone sparse sampling results obtained with SLADS with model training convergence on the left (a) and MSE/SSIM measurements on the right (b).

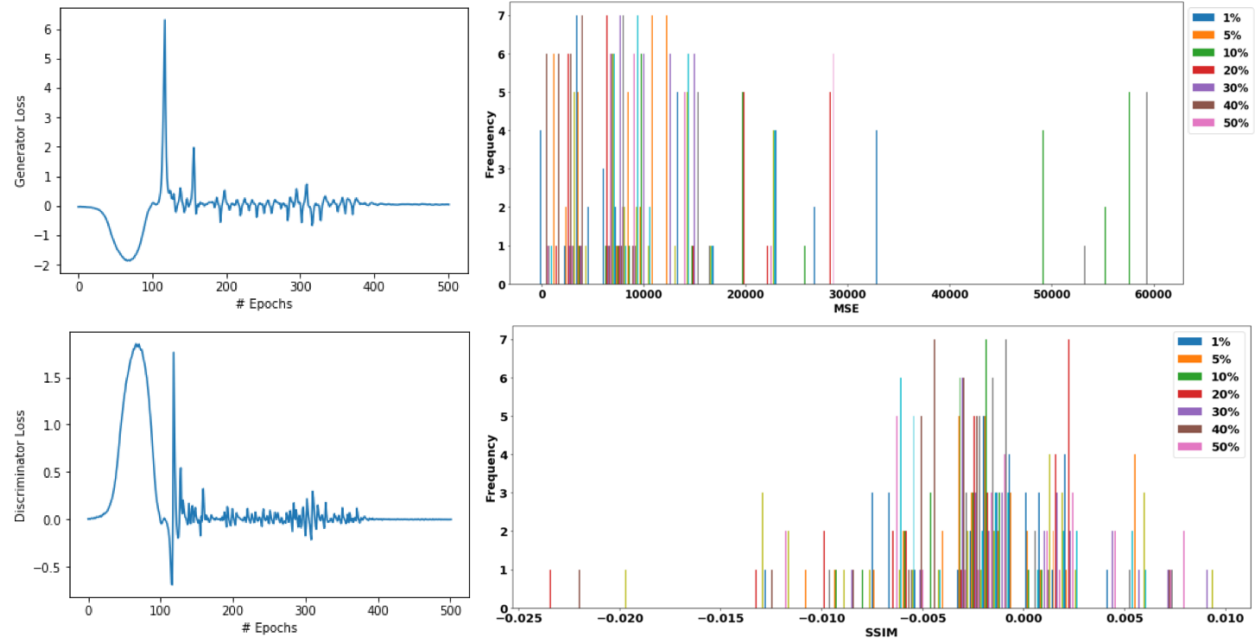


Figure 12: WDC-GLANDDS comparative performance on the sandstone dataset showing model losses on the left (a) and the MSE/SSIM results on the right (b).

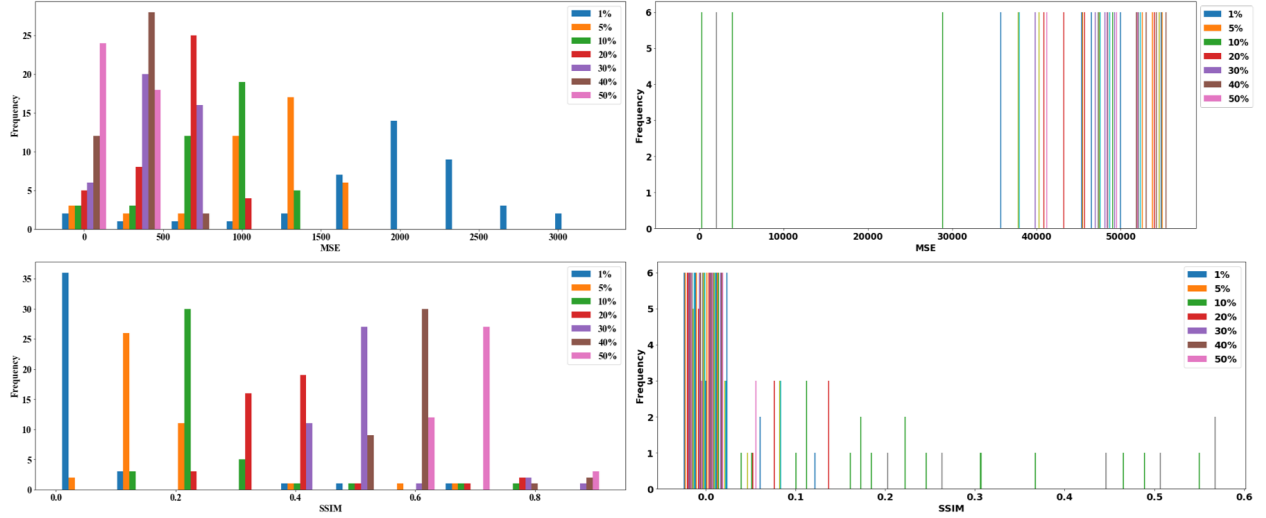


Figure 13: MSE/SSIM results of automatic thresholding as compared with manually created masks of the sandstone dataset as performed by SLADS on the left (a) and WDC-GLANDSS on the right

## Discussion

### SLADS

While SLADS was highly successful for both reconstruction of sparse sampled images and masking of non-homogeneous material, it still has some noteworthy issues. After training, SLADS chooses the best model ("theta" and "c" value pair) to perform reconstructions. However, since these models are conducted across a range of user specified starting sampling densities, it may be more appropriate to select the best model trained for the actually intended initial sampling density, rather than the best overall. Although not examined in detail, it seems likely that the reconstructions made from an initial 30% random sparse sampling would generally be superior to those made with less information. These later models should be better suited for implementation where only 1% initial sampling can be supplied, but the code has not yet implemented this selection process.

SLADS also remains based on a very simplistic linear model. Demonstrated in Figure 9 and 11, the model does not improve significantly with increasing numbers of training samples. Additionally, model performance peaked for both datasets at values above the

converged values, with only a couple of training images having been provided. This degree of overfitting could prove to be highly detrimental to the model’s ability to generalize, particularly worrisome in medical scanning contexts. If the scanning algorithm were to skip over areas of tissue it assumes to conform with learned patterns of idealized tissue, it could easily miss critical abnormalities. Applications that make use of SLADS should always fully scan areas identified as potentially containing information, rather than simply assuming it conforms to its predictions. Overall, this shows the limitation of using a linear model for the volume of training samples available for this and most machine learning research.

## **WDC-GLANDSS**

The WDC-GLANDSS model was ineffective at both reconstruction and masking tasks. Mode collapse was a highly common occurrence even with the mitigations provided through WGAN. The discriminator/generator losses became additionally increasingly variant with more epochs, despite aggressive levels of clipping applied to the discriminator. Particularly problematic was the establishment of hyperparameters suitable for the sandstone and pancreatic samples. The MNIST dataset contained roughly 4 times more samples, of which there were only 9 specific classes all perfectly centered in their images. The pancreatic samples were also centered, but were at a higher resolution set at 64x64 pixels. Using these as inputs to the model consistently caused mode collapses, or exponential gradients, eventually leading to the decision to downsample them to 28x28 pixels. While this did allow for the losses to roughly converge, mode collapse could not be avoided resulting in a failure to generalize sufficiently for any of the testing images. Even more of an issue was the sandstone which besides having 128x128 pixel resolution, has no specific classes, no contextual positioning information, widely varying pixel intensities, and altering sharpness levels. These could not even be downsampled to 28x28 due to the loss of any recognizable visual forms. Where the structures of handwriting formations and biological tissue have some degree of logical/consistent positioning, those of sandstone have been largely caused by entirely random

processes. Adding a gradient penalty to the WDC-GLANDSS model would theoretically remove the need to tune the discriminator weight clipping parameter. However, this was not allowable given the implementation time constraints, nor does the testing performed suggest that this would fully resolve the issues observed. Most likely to improve the model to a usable degree would be adjustments to the loss functions of both the GAN and secondary gradient descent performed for actual reconstruction.

The primary limitation of WDC-GLANDSS should be from its generalization of features, which seen in the MNIST results manifest in blurry images from the generator. SR (Super Resolution) NNs have been shown to be effective in upscaling low-resolution images.<sup>15</sup> It may be possible therefore, either through the integration of additional loss functions, or appending an additional NN stage to the architecture to produce much more realistic predictions.

## Conclusion

Medical sample acquisition time and thresholding of material grain structures for improved simulation and experimental workflows may be improved through the application of dynamic sparse sample reconstruction algorithms. SLADS has been shown in past work to significantly reduce the acquisition time for scanning medical images through application of a simple linear regression model. Combining this approach with a NN offers the potential to better generalize underlying structural information of training data and make use of higher quantities of training data.

Herein, SLADS has been shown to perform accurate sparse sample reconstructions for both pancreatic islets and sandstone structures. A combination of SLADS with DCGAN and WGAN architectures resulted in a new variant: WDC-GLANDSS, that was unable to match, or improve upon its linear counterpart. This was largely due to an inability to optimize hyperparameters and loss functions sufficiently for the datasets trained upon. Future work should continue to optimize these in order to overcome the linear model limitations observed

for SLADS. Given these limits, SLADS model’s performance must also continue to be more thoroughly evaluated for increased testing variations to ensure that critical information will not be overlooked during the reconstruction procedure.

## Formatting

This article is intended for publication in the Journal of Analytical Chemistry and therefore follows the guidelines/formatting outlined by the American Chemical Society (ACS).

## Supporting Information

Code for WDC-GLANDSS will be made available on a Github repository at:

<https://github.com/Yatagarasu50469/WDC-GLANDSS>.

## Acknowledgement

The author thanks Nathaniel Helminiak, from the Marquette University’s Opus College of Engineering, for providing the images of granular structure samples used herein and the Chemistry department, at Purdue University, for the use of the pancreatic islets dataset.

## References

- (1) Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. 2014.
- (2) Godaliyadda, G. M. D. P.; Ye, D. H.; Uchic, M. D.; Groeber, M. A.; Buzzard, G. T.; Bouman, C. A. *IEEE Transactions on Computational Imaging* **2018**, *4*, 1–16.
- (3) Scarborough, N. M.; Godaliyadda, G. M. D. P.; Ye, D. H.; Kissick, D. J.; Zhang, S.;



- Newman, J. A.; Sheedlo, M. J.; Chowdhury, A. U.; Fischetti, R. F.; Das, C.; Buzzard, G. T.; Bouman, C. A.; Simpson, G. J. *Journal of Synchrotron Radiation* **2017**, *24*, 188–195.
- (4) Zhang, S.; Song, Z.; Godaliyadda, G. M. D. P.; Ye, D. H.; Chowdhury, A. U.; Sengupta, A.; Buzzard, G. T.; Bouman, C. A.; Simpson, G. J. *Analytical Chemistry* **2018**, *90*, 4461–4469.
- (5) Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. 2015.
- (6) Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. Proceedings of the 34th International Conference on Machine Learning. International Convention Centre, Sydney, Australia, 2017; pp 214–223.
- (7) Wolterink, J. M.; Kamnitsas, K.; Ledig, C.; Igum, I. Generative adversarial networks and adversarial methods in biomedical image analysis. 2018.
- (8) Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. C. *CoRR* **2017**, *abs/1704.00028*.
- (9) Wolterink, J. M.; Leiner, T.; Viergever, M. A.; Igum, I. *IEEE Transactions on Medical Imaging* **2017**, *36*, 2536–2545.
- (10) Nie, D.; Trullo, R.; Lian, J.; Wang, L.; Petitjean, C.; Ruan, S.; Wang, Q.; Shen, D. *IEEE Transactions on Biomedical Engineering* **2018**, *65*, 2720–2730.
- (11) Schlegl, T.; Seeböck, P.; Waldstein, S. M.; Schmidt-Erfurth, U.; Langs, G. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. Information Processing in Medical Imaging. Cham, 2017; pp 146–157.
- (12) Kim, K. H.; Do, W.-J.; Park, S.-H. *Medical Physics* **2018**, *45*, 3120–3131.

- (13) Sankur, B. *Journal of Electronic Imaging* **2004**, *13*, 146.
- (14) LeCun, Y.; Cortes, C. **2010**,
- (15) Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.;  
Tejani, A.; Totz, J.; Wang, Z.; Shi, W. **2017**,