

Unpaired Magnetic Resonance Image-to-Image Translation for Prostate using Cycle-Consistent Adversarial Networks

Steve Burke
Stanford University
sfburke@stanford.edu

Vidush Mukund
Stanford University
vmukund@stanford.edu

Brian Wilcox
Stanford University
wilcoxee@stanford.edu

Abstract

In this paper we discuss an approach to transform apparent diffusion coefficient (ADC) magnetic resonance images to/from T2-weighted images via a set of generative adversarial networks. This problem is motivated by the need to reduce the long scan times to acquire MR images while still providing the information necessary for clinicians to make appropriate diagnoses. We explore the current state-of-the-art GAN models from inception to similar medical applications as well as suitable evaluation metrics. We then describe the dataset of images acquired from ProstateX and describe two major methodologies used for this modeling, CycleGAN and DualGAN. From these models, we collect generated images and perform quantitative and qualitative analysis. Finally, we determine that CycleGAN produces crisper and more realistic MR images and discuss plans for extending the research.

1. Introduction

Early detection of prostate cancer has a significant impact on the prognosis for patients. Histologies, which involve taking samples of tissue, can be high cost and high risk for patients, so MRI scans are useful for clinicians to first determine whether a patient needs a biopsy. MR image acquisition typically takes forty-five minutes and involves taking both T2-weighted images and ADC images, which each provide different structural information for analysis. We aim to reduce the scan time for acquiring MR images as well as the number of scans needed per patient, while still providing clinicians with all of the information needed for an accurate diagnosis.

Generative adversarial networks (GANs) have been used to effectively map between different image modalities [1]. By training the network on unpaired sets of data of the two image types, the network can then take an image of one type as input and output a corresponding image of the second image type. We will apply this approach to the analysis of MR

images, and attempt to map between the T2-weighted and ADC image modalities. This method could allow a clinician to acquire a single scan type from a patient and then use a trained GAN network to produce a corresponding image of the other scan type.

The dataset for this task is a set of publicly available MR images taken for diagnosis of prostate cancer [4]. The GAN architectures generate images of the complementary modality ($ADC \rightarrow T2$ or $T2 \rightarrow ADC$), but they also generate images of the same modality as the input image ($ADC \rightarrow T2 \rightarrow ADC$, for example). While generated images that are the same modality as the input MRI scan would not provide any additional information for medical diagnosis, we compare these generated images to the original real image data as confirmation that the network has trained effectively.

We evaluate the performance of our experimental GAN architectures by qualitatively examining the generated images for any discernible differences between generated and original images for each modality. For certain images in our data set that comes in pairs, we are able to compare the generated images to the original paired test image as well. Additionally, we use a nearest neighbor comparison of generated images to training images to check for overfitting to the training data, and reconstruction distance measurements to quantify the performance of each GAN.

2. Related Work

Goodfellow et al. first suggested the application of a generator-discriminator model for estimating generative models through an adversarial network [5]. This framework has expanded to several innovations including the Least-Squares GAN (LSGAN) model proposed in Mao et al. [6]. This model implements a least squares loss function for the discriminator, to prevent vanishing gradients in the network and achieve higher quality results. Another major development in the field of GANs was the innovation of Deep Convolutional GAN (DCGAN) by Radford et al. which led to significantly improved results in GAN model outputs [7]. These deeper models serve as one of the best general mod-

els for image representations. Notably, one of the defining characteristics is the paired mappings from distribution A to distribution B. In fact, there has been a lot of work done in the field related to paired image generation [8] [9] [10]. None of these, however, take advantage of the concept of cycle-consistency.

Along with significant progress in the field of paired image generation, there have been significant strides in the development of unsupervised GAN methodologies. In particular, Liu et al. implemented an unsupervised model for image-to-image translation by learning a joint distribution of images from a marginal distribution of an individual domain [11]. In order to infer these distributions, Liu implements a latent feature space using a Coupled GAN framework. In the field of medical imaging, there have been several attempts to generate GANs with the goal of being able to synthesize more medical data for various applications. Some of these applications include trying to generate more scans of specific organs [12], while others focus on trying to generate transformations from MRI to Computed Tomography (CT) scans for generating cancer images [13] [14]. The results of the mappings for the MRI to CT results have the advantage of well-defined statistical distributions that are different but stylistically distinct. This is not the case with our dataset.

There are a number of methods described in the literature for evaluating GANs qualitatively and quantitatively, but no current consensus on the best metrics [15]. For qualitative analysis of GAN results, it is common to compare samples of generated images to their nearest neighbors from the training set, in order to detect overfitting to the training data [15] [16]. For quantitative analysis, we examine the reconstruction loss for the data, which is indicative of which mappings have a one-to-one correspondence. This is useful in discerning whether the generators preserve the statistical properties of the corresponding images.

For quantitative analysis of GANs, it is also common to use Inception or Mode Score criteria [17]. However, we chose to use the reconstruction error analysis instead, because these methods fail to address our particular application for GANs. A pre-trained model would not contain MRI prostate scans. This would require training on an Inception architecture for these MRI scans to classify between T2-weighted and ADC. There is no guarantee that this would be a telling metric for our dataset (which is full of images in both distributions that can vary significantly).

3. Dataset and Features

The MRI scans were originally stored as DICOM (Digital Imaging and Communications in Medicine) images [4]. Since raw DICOM image pixel intensity values vary from machine to machine, all images were normalized between 0 and 1, then scaled to the PNG range of 0 to 255, and finally

stored as PNG images.

To correct for different image sizes and fields of view for the two contrast types, ADC and T2-weighted, the sections of the transaxial image slices containing the prostate, bladder, and colon were cropped and resized to a uniform 256x256 pixels for all images. The data was then split into train and test sets. In total there were 345 patients with at least 18 T2-weighted and 18 ADC images each. 300 patients' scans were grouped as the train data and 45 patients' scans were grouped as the test data.

The train set was augmented by flipping the images over the vertical axis. This augmentation was primarily done to ensure that the networks learned that the axial slices of the prostate have weak symmetry over the vertical axis.

To ensure that all images in the T2 set had corresponding images in the ADC set and vice versa for each patient, images without corresponding pairs were removed from the dataset. This pairing of images allows for comparison of original scans with generated scans of the same type of contrast.

The training set consisted of 5,827 T2-weighted scans and 5,827 ADC scans, which were then augmented, as described above, to 11,654 images in each modality for a combined total of 23,308 images. The test set consisted of 864 images in the T2-weighted modality and 864 images in the ADC modality for a total of 1,728 images.

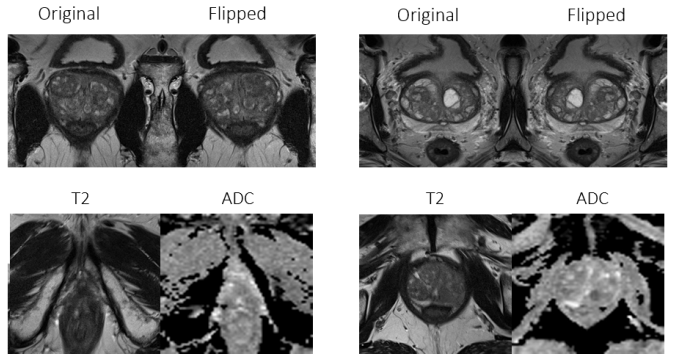


Figure 1. This figure shows some representative scans from the dataset. The first row of the figure shows the data augmentation via flipping the original images across the vertical axis. The second row shows two sets of paired images, T2 and ADC contrast scans.

4. Methods

The goal of our models is to learn two mapping functions between the T2 image domain and the ADC image domain. The first architecture we discuss is the CycleGAN architecture proposed by Zhu et al. 2017 [1]. The second architecture we discuss is the DualGAN architecture first proposed by Yi et al. 2017 [2].

4.1. Architectures

4.1.1 CycleGANs

The cycle-consistent adversarial network, or CycleGANs architecture, models the functional mappings via two generators and two discriminators. One generator for each mapping and one discriminator for each generator. The discriminator is a 5 layer convolutional neural network (CNN), and the generator uses a residual network (ResNet) with 2 convolutional layers, 9 residual blocks, and 2 transpose convolutional layers. The discriminators D_X and D_Y attempt to distinguish between real and generated images. The generators, meanwhile, generate images similar to scans from the respective image domain in an attempt to fool the paired discriminators.

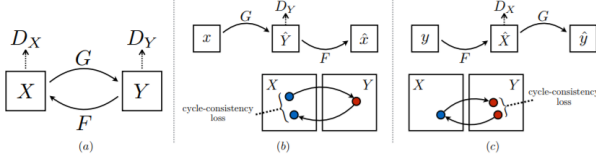


Figure 2. a) The CycleGANs architecture constructs two mappings, $G: X \Rightarrow Y$ and $F: Y \Rightarrow X$. The discriminators are D_X and D_Y respectively. b,c) Two cycle-consistency losses are introduced to ensure that translation from one domain to another is invertible, i.e. $F(G(x)) \approx x$ and $G(F(y)) \approx y$. [1]

In this paper, we denote the data distributions for the two image domains as $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$. We imagine the training samples as sampled from these distributions. The functions F and G are defined as the mappings between these two distributions. We also define the training samples as $(x_i)_{i=1}^N$ where $x \in X$ and $(y_i)_{i=1}^N$ where $y \in Y$. For the mapping function $G: X \Rightarrow Y$ and its associated discriminator D_Y , we used the losses as expressed in Figure 3.

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)} [\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)} [\log(1 - D_Y(G(x)))]$$

Figure 3. GAN loss used by CycleGAN

The mapping function $F: Y \Rightarrow X$ has a similar adversarial loss associated with it. An additional cycle-consistency loss is defined to ensure the two mapping that are learned do not contradict one another. This additional loss function also reduces the space for mapping functions by imposing the additional constraint of invertibility. This loss function enforces that mappings F and G have the properties $F(G(x)) \approx x$ and $G(F(y)) \approx y$.

The final objective function with both losses is shown in Figure 5. The lambda term acts as a control for the relative importance of the cycle-consistency loss.

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)} [\|G(F(y)) - y\|_1]$$

Figure 4. To incentivize cycle-consistent behaviour and to prevent the two mapping from contradicting each other, we used the cycle-consistent loss as defined in Zhu et al.[1]

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F)$$

Figure 5. Full objective function for CycleGAN

The final optimization problem our model aims to solve is described in Figure 6. The generators attempt to minimize the objective function, while the discriminators attempt to maximize the objective function as captured in the optimization problem.

$$G^*, F^* = \arg \min_{G, F} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y)$$

Figure 6. Optimization problem for CycleGAN

4.1.2 DualGANs

Similarly to CycleGANs, this model constructs a mapping of A to B and B to A to obtain a primal-dual mapping between the two outputs. The model trains for the primal and dual mapping simultaneously, just as CycleGANs does, to allow for an informative feedback loop. Both architectures use a 5 layer CNN to produce generated outputs. One of the key differences between this architecture and the CycleGANs architecture is the use of a U-Net structure for the Generator Architecture [3]. The architecture is shown in Figure 7.

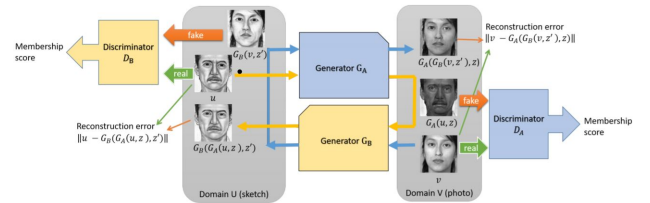


Figure 7. The DualGAN architecture constructs two mappings, $G_B(A)$ to B and $G_A(B)$ to A. The discriminators are D_B and D_A respectively. Two reconstruction losses are introduced to ensure that translation from one domain to another is invertible, i.e. $G_A(G_B(A)) \approx A$ and $G_B(G_A(B)) \approx B$. These loss functions are shown in Figures 8 and 9 respectively [2].

The loss is derived in terms of the discriminator and generator as explained in Yi et. al. The discriminator loss

follows the format demonstrated in the Wasserstein GAN (WGAN) [18]:

$$\begin{aligned} l_A^d(u, v) &= D_A(G_A(u, z)) - D_A(v), \\ l_B^d(u, v) &= D_B(G_B(v, z')) - D_B(u), \end{aligned}$$

Figure 8. DualGAN Discriminator Loss Function

$$\begin{aligned} l^g(u, v) &= \lambda_U \|u - G_B(G_A(u, z), z')\| + \\ &\quad \lambda_V \|v - G_A(G_B(v, z'), z)\| \\ &\quad - D_A(G_B(v, z')) - D_B(G_A(u, z)), \end{aligned}$$

Figure 9. DualGAN Generator Loss Function

One of the major benefits of the U-Net based architecture in the generator model is the ability to share low-level information between the input and the output layers. Instead of applying the noise vectors shown in the generator loss function, they are supplemented in the form of dropout in several layers.

The discriminator models employ a Markovian PatchGAN, which assumes independence between pixels distanced beyond a specified patch region. This is helpful for capturing local high-frequency features like texture and style. This helps in ensuring that the recovery loss encourages preservation of global and local frequency information, which prevents overfitting in the model and is ideal for a medical image dataset.

The training of the model follows that of the WGAN. The discriminators are trained for a specified number of steps and then one step on the generator is taken. In training the discriminator model, weights are updated for a specified number of iterations and then the generator model is updated. To keep weights stable, clipping is applied for each update. Once the generator takes a sample, weights are then updated to minimize the generator loss function. In traditional GAN networks, when the discriminator improves, the sigmoid cross-entropy loss is saturated and can lead to vanishing gradient problems. The training algorithm is shown in Figure 10.

One of the benefits of the Wasserstein method used in this training set is that the loss function is differentiable almost everywhere, with the result that the learning scheme produces a better discriminator in the overall objective function.

In running the DualGAN, it was noted that the preferred loss metric for calculating the distance between two images is L_1 as opposed to the L_2 distance function used in the CycleGAN code. For the sake of completeness, both distance functions were used in training the models and similar hyperparameters were used for the L_2 run in comparison to

Algorithm 1 DualGAN training procedure

Require: Image set U , image set V , GAN A with generator parameters θ_A and discriminator parameters ω_A , GAN B with generator parameters θ_B and discriminator parameters ω_B , clipping parameter c , batch size m , and n_{critic}

```

1: Randomly initialize  $\omega_i, \theta_i, i \in \{A, B\}$ 
2: repeat
3:   for  $t = 1, \dots, n_{critic}$  do
4:     sample images  $\{u^{(k)}\}_{k=1}^m \subseteq U, \{v^{(k)}\}_{k=1}^m \subseteq V$ 
5:     update  $\omega_A$  to minimize  $\frac{1}{m} \sum_{k=1}^m l_A^d(u^{(k)}, v^{(k)})$ 
6:     update  $\omega_B$  to minimize  $\frac{1}{m} \sum_{k=1}^m l_B^d(u^{(k)}, v^{(k)})$ 
7:      $clip(\omega_A, -c, c), clip(\omega_B, -c, c)$ 
8:   end for
9:   sample images  $\{u^{(k)}\}_{k=1}^m \subseteq U, \{v^{(k)}\}_{k=1}^m \subseteq V$ 
10:  update  $\theta_A, \theta_B$  to minimize  $\frac{1}{m} \sum_{k=1}^m l^g(u^{(k)}, v^{(k)})$ 
11: until convergence

```

Figure 10. DualGAN Training Algorithm using Wasserstein method

the hyperparameters used in the CycleGAN code. In both images, data augmentation was implemented by flipping the images horizontally. This maintains the realistic characteristics of the prostate scans while increasing the amount of data present. The first run used unpaired data in the training and test set. The second run used paired data for the T2-weighted and ADC images, so that there was a guarantee of having a corresponding real image for baseline comparison.

5. Results

5.1. Preliminary CycleGANs Qualitative Results

Below are a few preliminary result images from our initial testing with a CycleGANs network trained over 300 patients for 50 epochs. This testing used an Adam optimization method with a learning rate of $2e^{-4}$.

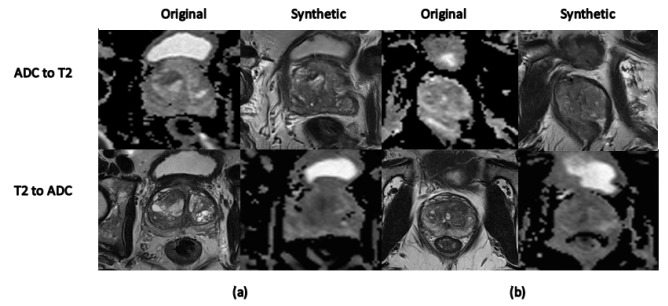


Figure 11. Preliminary CycleGAN Results: a) The four images in the left 2x2 block correspond to patient a. The first column has an ADC scan and its paired T2-weighted scan, while the second column has the corresponding generated images from the CycleGAN network. The right 2x2 block has the same representation for patient b.

From these preliminary results, we can see that training on unpaired images in this unsupervised approach gener-

ated images that are very similar to the seed images but not similar to the corresponding ground-truth scan pair, i.e. the synthetic ADC image is very similar to the seed T2 image but not to the corresponding real ADC scan. These differences may be due to the different conditions in which these MRI scans were generated. For example, the patient may have moved between scans, or the scan coordinates in the z direction may be slightly offset by some delta between scans.

5.2. Preliminary DualGAN Qualitative Results

Results for initial DualGAN testing are below. Specifically, the network successfully reconstructs the original input images by taking the inverse mapping of each transformed image ($G_B(G_A(B))$). The optimization algorithm used was Adam with a learning rate of $5e^{-5}$ and $\beta_1 = 0.5$. Lambda parameters of 20.0 for ADC and T2-weighted, as well as a batch size of 64 over 45 epochs were used for training this network.

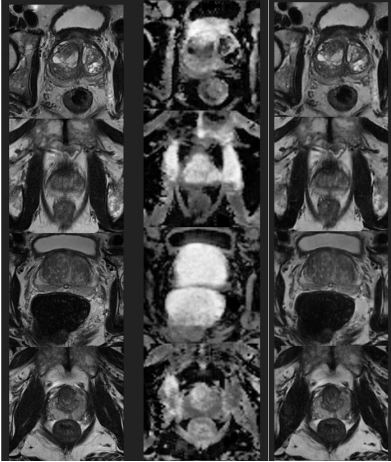


Figure 12. Preliminary DualGAN: T2 to ADC to T2 results: From left to right, one can visualize a patient’s scan in T2-weighted format transformed to ADC and then back to T2-weighted.

From these results, we can see that training on unpaired images has led to strong reconstructions in going from ADC images to T2-weighted images. However, the results going from the T2-weighted to ADC are not as successful, as the organs of interest are not colored as expected in realistic ADC images.

5.3. Final Results for CycleGANs and DualGANs

Figures 14 and 15 demonstrate examples of generated ADC and T2 images for each of three final GAN architectures that we chose after optimizing over the hyperparameters. The three final GANs, one CycleGAN and two DualGANs are detailed below.

For our CycleGAN model, we set $\lambda = 10$ for the equation in Figure 5 and used the Adam optimizer with a batch

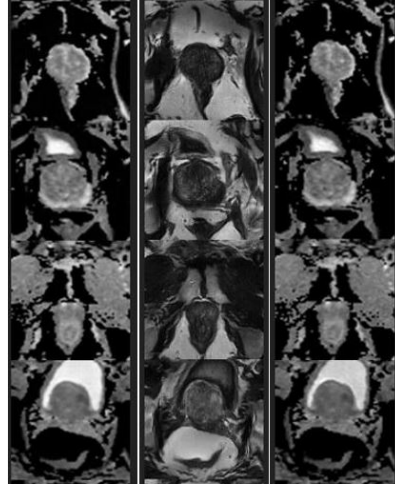


Figure 13. Preliminary DualGAN ADC to T2 to ADC Results: From left to right, one can visualize a patients’ scan in ADC Format transformed to T2-weighted and then back to ADC.

size of 1, a learning rate of $2e^{-4}$ with a decay every 10 epochs, and a beta1 of 0.5. The model was trained for 50 epochs. The loss function optimized was the LSGAN loss.

For our DualGAN Model 1, we used RMSprop with a batch size of 64, a learning rate of $5e^{-5}$ with decay every 3 epochs, and a beta1 of 0.5. DualGAN Model 1 was trained for 50 epochs and we set $\lambda = 20$ for both datasets. The loss function optimized for Model 1 was the L1 loss.

For our DualGAN Model 2, we used RMSprop with a batch size of 64, a learning rate of $2e^{-4}$ with decay every 10 epochs, and a beta1 of 0.5. DualGAN Model 2 was trained for 50 epochs and we set $\lambda = 10$ for both datasets. The loss function optimized for Model 2 was the L2 loss.

6. Discussion

6.1. Nearest Neighbor Qualitative Analysis

To detect overfitting to the training data for each of our three final GANs, we compare five random samples of generated ADC and T2 images to their L2 distance nearest neighbors from the training data set. The nearest neighbors are compared side by side to the generated images in Figures 16 and 17. We also found the nearest neighbor for each generated image using L1 distance, but did not include these images in the figures because in almost every case, the images were identical for L1 and L2.

Note that the same five randomly selected test images were used for selecting the corresponding 5 generated images displayed for each architecture. This allows for a comparison of generated images across architectures by comparing the generated images across each row in the figures, with the expectation that the generated images should look somewhat similar.

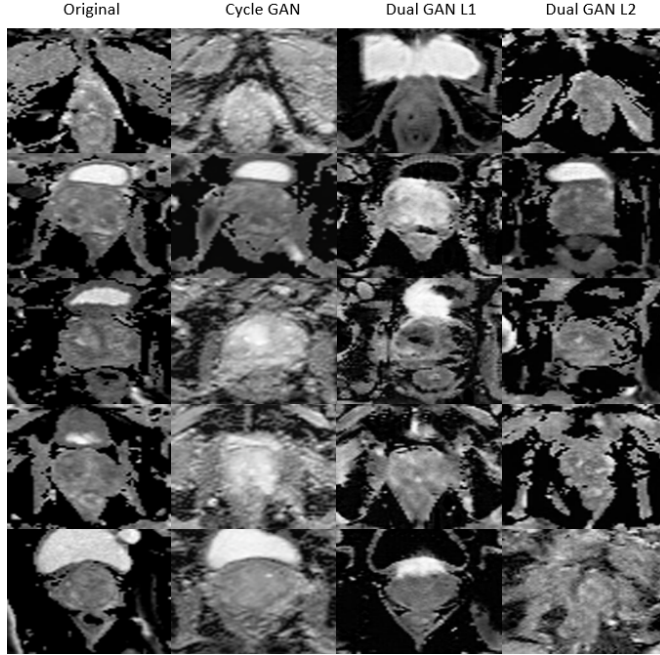


Figure 14. The second, third, and fourth columns above show generated ADC images for each of our three final GANs. The first column shows the ADC test images which are the corresponding pairs of the T2 test images that were input to the GANs.

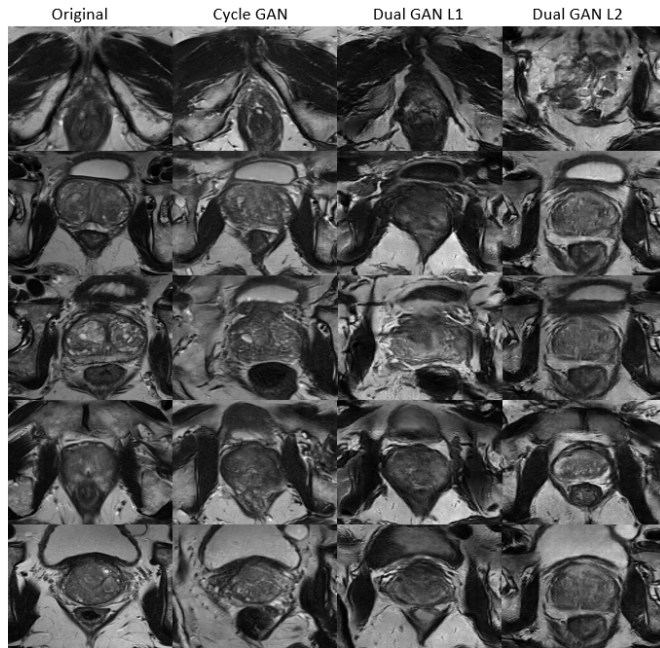


Figure 15. The second, third, and fourth columns above show generated T2 images for each of our three final GANs. The first column shows the T2 test images which are the corresponding pairs of the ADC test images that were input to the GANs.

Both figures show that the nearest neighbors found for the generated images do not generally look very similar to

the generated images. This indicates that the models were not overfitting to the training data for either ADC or T2 image generation. However, in many cases the same training image was identified as the nearest neighbor for multiple different generated images, especially for the ADC images. This indicates that there may be limitations on how effective this nearest neighbor analysis is for our particular dataset.

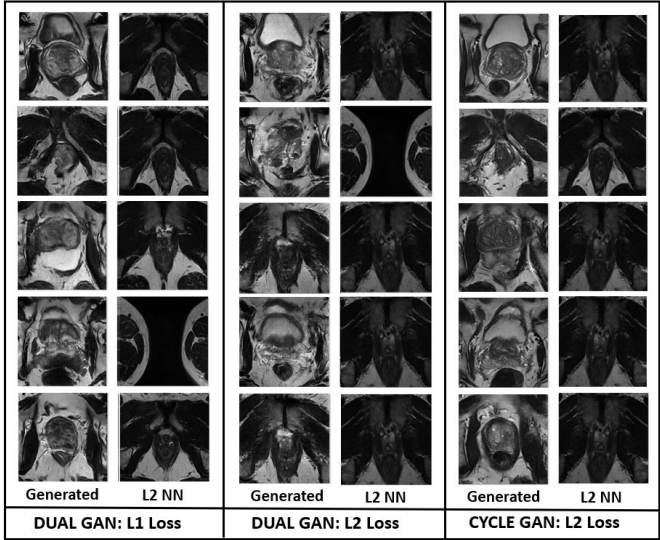


Figure 16. For each of the three GAN architectures, five random samples of T2 generated images are shown above with their corresponding L2 Nearest Neighbor from the T2 training set.

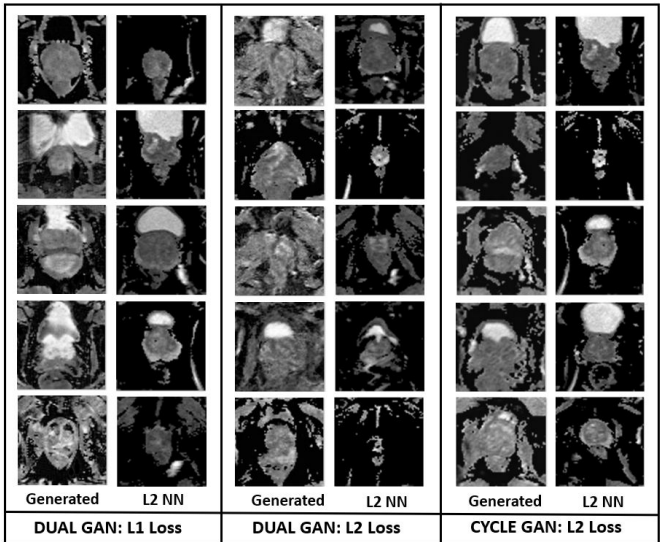


Figure 17. For each of the three GAN architectures, five random samples of ADC generated images are shown above with their corresponding L2 Nearest Neighbor from the ADC training set.

6.2. Reconstruction Distance Quantitative Analysis

Figures 18 and 19 represent the measured average L1 and L2 distances between generated images of each type and their corresponding images from the test data for the three GANs. For the single direction cases (e.g. T2 to ADC), the distance is measured between the generated image and the test image of the same modality for all available cases where the test data had a pair of images (one T2 and one ADC). For the dual direction cases (e.g. T2 to ADC to T2), the distance is measured between all generated images and their corresponding original test images that were input to the GAN.

The distance results show that in general the distances for the single direction cases are lower than for the dual direction cases. This is somewhat surprising, as we might expect the distance to be lower for the dual direction case, as the GAN should be able to effectively map back to the original modality that was input to the network. Additionally, the distances for the DualGAN cases appear to be slightly lower than those for the CycleGAN, although we do not have data on the dual cases for the CycleGAN.

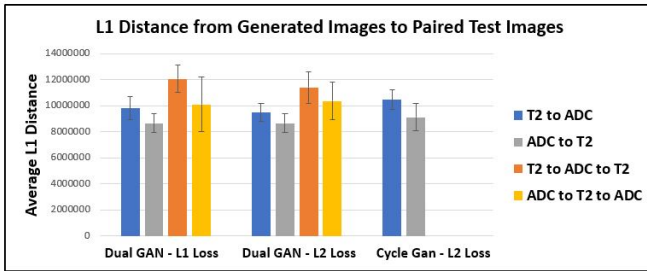


Figure 18. Average L1 distance between generated images and corresponding images from the test data set for three GAN architectures. Single mapping cases compare generated images to paired images from the test data set. Dual mapping cases compare generated images to original input test images of the same modality.

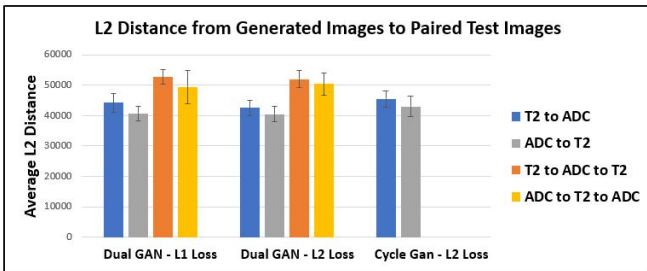


Figure 19. Average L2 distance between generated images and corresponding images from the test data set for three GAN architectures. Single mapping cases compare generated images to paired images from the test data set. Dual mapping cases compare generated images to original input test images of the same modality.

7. Conclusion/Future Work

While the L1 and L2 reconstruction losses are lower for the DualGAN, from a qualitative perspective the CycleGAN model produce images most similar to ground truth images. The CycleGAN algorithm may have performed better since the Wasserstein loss function used by the DualGAN model trains the discriminators for multiple steps before training the generator. Though this speeds up training time for the model, it comes at the cost of more realistic MR images.

For the results shown above, a ResNet architecture was used for the generators in the CycleGAN model. Moving forward, we are looking into using a U-Net architecture for the generators and preliminary results are promising.

The Prostatex dataset contains some scans that do not have the prostate present. For a future iteration, the dataset should be cleaned of any images that do not have the prostate in the field of view. This should ensure that the models develop features tied specifically to prostate.

For further result analysis in the future, we will seek out the opinions of medical experts to see if they are able to discriminate between real image scans and our generated images, and whether the generated images can relay the same amount and quality of information for diagnosis as the ground truth scans.

8. Acknowledgements

Steve wrote scripts for data augmentation as well as for collecting qualitative and quantitative results on the output of each GAN after researching common methods for GAN analysis. Specifically, the scripts added flipped images to the original data set, found the nearest neighbors of generated images from the corresponding training data sets, and calculated the L1 and L2 distances between generated images and corresponding test images.

Brian worked on implementing the DualGAN and generating various runs for the results section of the paper. Additionally, he helped in formulating which distance metrics to use for comparing the results of GANs, as well as identifying comparable objectives with medical GANs in the literature.

Vidush wrote the scripts that extracted prostate images from DICOM files, cropped the images, and normalized the images. Vidush also worked on implementing the CycleGAN model, training a ResNet and a U-Net architecture, and contributing to the results for the CycleGAN model.

This project was done in collaboration with the Laboratory for Integrative Personalized Medicine (PIMed) which is directed by Dr. Mirabela Rusu, PhD. Models were trained on the GPU nodes of the Sherlock Cluster with access provided by PIMed.

CycleGANs code was based on an implementation of the

model described in Zhu et al. 2017 [1] by Xiaowei Hu [19].
DualGAN code was based on an implementation of the model described in by Z. Yi et al. 2017 [2] by Jack Yi [21]

References

- [1] Jun-Yan Zhu*, Taesung Park*, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", in IEEE International Conference on Computer Vision (ICCV), 2017.
- [2] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong "DualGAN: Unsupervised Dual Learning for Image-to-Image Translation", in IEEE International Conference on Computer Vision (ICCV), 2017.
- [3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox "U-Net: Convolutional Networks for Biomedical Image Segmentation", in Medical Image Computing & Computer Assisted Intervention (MICCAI), 2015.
- [4] Prostatex data: <https://wiki.cancerimagingarchive.net/pluginservlet/mobile?contentId=23691656#content/view/23691656>
- [5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In Proceedings of NIPS, pages 2672-2680, 2014.
- [6] X Mao, Q Li, H Xie, R YK Lau, and Z Wang. Multiclass generative adversarial networks with the l2 loss function . 2016.
- [7] A. Radford, L. Metz, Unsupervised representation learning with Deep Convolutional Generative Adversarial Networks, 2016.
- [8] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In NIPS, pages 4694-77, 2016.
- [9] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. ECCV, 2016.
- [10] T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixelwise correspondences. In CVPR, pages 1191-1200, 2015.
- [11] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. arXiv preprint arXiv:1703.00848, 2017.
- [12] J. T. Guibas, T. S. Virdi, and P. S. Li. (Dec. 2017). Synthetic medical images from dual generative adversarial networks. [Online]. Available: <https://arxiv.org/abs/1709.01872>
- [13] D. Nie et al., Medical Image Synthesis with Deep Convolutional Adversarial Networks, in IEEE Transactions on Biomedical Engineering. doi: 10.1109/TBME.2018.2814538
- [14] D. Nie, R. Trullo, C. Petitjean, S. Ruan, and D. Shen. (2016). Medical image synthesis with context-aware generative adversarial networks. [Online]. Available: <https://arxiv.org/abs/1612.05362>
- [15] Ali Borji. Pros and cons of GAN evaluation measures. arXiv preprint arXiv:1802.03446, 2018
- [16] Theis, L., van den Oord, A., and Bethge, M. A note on the evaluation of generative models. arXiv:1511.01844, Nov 2015. URL <http://arxiv.org/abs/1511.01844>.
- [17] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. arXiv preprint arXiv:1612.02136, 2016.
- [18] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017.
- [19] CycleGAN code: <https://github.com/XHUIJOY/CycleGAN-tensorflow>
- [20] DualGAN code: <https://github.com/duxingren14>
- [21] Libraries: OpenCV, TensorFlow, SciPy