Ontario Tech University
Faculty of Science, Computer Science

Project Group: 22 April 6, 2024

Computer Vision Course Project: Unpaired Image-to-Image Translation Using CycleGANs for CT and MRI Brain Scans

by

Aaveg Shangari Srivathsan Sivakumar

Abstract:

CT scans expose patients to ionizing radiation. MRI scans are physically restrictive and the patient needs to endure loud noise for 20 to 40 minutes. CycleGANs are a siamese-model of Generative Adversarial Networks which aims to reduce the necessity for multiple scanning modalities in medical diagnostics, potentially alleviating patient discomfort and exposure risks. This technology can potentially assist medical practitioners with diagnosis and prediction of brain turmors. Unpaired brain scan images from Kaggle were used. The GAN was trained with Adam optimizer with Beta1 and Beta2 values of 0.5 and 0.999 respectively. The Cycle Consistency Loss weight was set to $\lambda = 10$. The learning rate was set to 0.0002 and the model was trained for 10 epochs. The results proved the effectiveness of the CycleGAN. Training for more epochs and optimizing using cutting-edge frameworks is expected to significantly increase the performance of the model.

Keywords: Generative Adversarial Networks, CycleGAN, PyTorch, CT, MRI, Deep Learning

1 Introduction

1.1 Motivation

Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) are commonly used in the diagnosis of brain tumors. In some cases, to reinforce a diagnosis, both techniques are used. This situation will prove to be uncomfortable for patients because CT scans expose them to ionizing radiation, and MRI scans require them to be completely still and endure loud noise for 20 to 40 minutes [1]. The limitations of each scan technique can be overcome using Cycle Generative Adversarial Networks (CycleGANs) since they are able to perform style transfer and generate corresponding images [1].

1.2 Cycle Generative Adversarial Networks

A CycleGAN is a neural network used for image-to-image translations without paired examples. Traditional methods depend on having two corresponding images, such as a photo of a horse next to a photo of the same horse as a zebra. A CycleGAN, however, can learn this translation with unpaired images from different domains[2].

It employs generators and discriminators. Generators perform the translation task, converting an image from one domain (like photos of horses) to another (like photos of zebras), and vice versa. Discriminators then evaluate the authenticity of the translated images, challenging the generators to produce more convincing results[2].

The ”cycle” in CycleGAN describes the translation of an image to a new domain and back to the original, with the aim of retaining the original image’s essence. This capability is useful for applications such as style transfer, season change in landscapes, object transformations in images, and photo quality enhancement. In these processes, the generators and discriminators ensure that the final images are realistic and maintain the integrity of the original domain.

Objective Function:

The total loss function for CycleGAN is given by:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F), \quad (1)$$

where $\mathcal{L}_{GAN}$ represents the adversarial loss for each domain, and $\mathcal{L}_{cyc}$ is the cycle consistency loss. The parameter λ determines the relative weight of the cycle consistency loss [2].

Adversarial Loss:

For the generator G and discriminator D_Y, the adversarial loss is defined as:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F), \quad (2)$$

which calculates the expectation of the log probability that D_Y correctly classifies real images y and the log probability that D_Y is fooled by the fake images generated by G [2].

For the generator F and discriminator D_X, the adversarial loss is similarly calculated with roles reversed [2].

Cycle Consistency Loss:

Cycle consistency loss ensures translations are consistent in both directions:

$$\mathcal{L}_{\mathrm{cyc}}(G, F) = \mathbf{E}_{x \sim p_{\mathrm{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbf{E}_{y \sim p_{\mathrm{data}}(y)}[\|G(F(y)) - y\|_1], \quad (3)$$

where $\|\cdot\|_1$ denotes the L1 norm. This loss ensures that an image x from domain X, after being transformed to domain Y and back again, should resemble the original image x. The same applies for images from domain Y [2].

In summary, CycleGAN trains to achieve two objectives: (1) translate images between domains to fool the discriminator (adversarial loss), and (2) maintain the identity of images through the translation cycle (cycle consistency loss). These combined losses enable the CycleGAN to learn meaningful transformations between two unpaired image domains.

1.3 Dataset and Pre-processing

The dataset used in this study consists of brain scan images derived from Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) sources, which are available on the Kaggle website. These images are intended for the application of unpaired image-to-image translation. The dataset is categorized into two domains: Domain A, which includes CT brain scan images, all formatted to a uniform resolution, and Domain B, containing MRI brain scan images that vary in size. Given the variability in Domain B, image resizing is an important preprocessing step to ensure uniformity across the dataset.

To further prepare the images for training, the Albumentations library is utilized to compose a sequence of transformations. This compositional approach is employed to resize the images to a uniform resolution of 256×256 pixels and to normalize the pixel values to fall within the range of -1 to 1. Following the normalization, the images are converted into tensors using a ToTensorV2 transformation. This ensures a standardized input format for the CycleGAN model, which is critical for the consistency and effectiveness of the unpaired image-to-image translation tasks [2].

The described preprocessing steps ensure that the images are suitably prepared for their use in a CycleGAN model, which is particularly effective for the task of unpaired image-to-image translation. By adhering to this preprocessing regimen, the dataset is optimized for engaging with the complexities involved in training state-of-the-art machine learning algorithms for medical image analysis.

1.4 Model Architecture

A visual representation of the generation architecture can be found in Figure 1. The construction of each generator is characterized by a distinct configuration that encompasses two downsampling layers, a suite of residual blocks—integral for supporting deep network training by addressing the vanishing gradient dilemma—and a duo of upsampling layers dedicated to image reconstruction. Adhering to established practices, six residual blocks were typically incorporated for processing smaller images, with the count escalating to nine for images of larger dimensions [2].

As proposed in Figure 2, the discriminators Dx and Dy are structured upon a PatchGAN design. This approach features a cascade of convolutional layers that are adept at discerning and classifying sub-regions, or 'patches', of an image as authentic or synthetic, thus sharpening the model's acuity for detailed features. The architecture customarily initiates with 64 filters and proceeds to double this number in each succeeding layer, a convention notated as Ck to symbolize a Convolution-InstanceNorm-LeakyReLU layer [2].

For normalization, instance normalization was the preferred choice over batch normalization, optimizing the input standardization across each channel for every individual training sample.

The network's non-linear responsiveness is orchestrated through the implementation of ReLU activation functions in the generator, contrasted by LeakyReLU in the discriminator. The LeakyReLU's design permits a slight gradient when inactive, a feature beneficial for the sustenance of gradient flow throughout the training phase.

Beyond the standard adversarial loss that is a hallmark of GANs, prompting generators to produce images that seamlessly blend with authentic images from the target domain, the CycleGAN is further refined with a cycle consistency loss. This additional loss criterion is pivotal, ensuring that images that undergo a domain transfer and then revert to their original domain bear a strong resemblance to their initial form, thereby upholding a logical and coherent inter-domain translation.

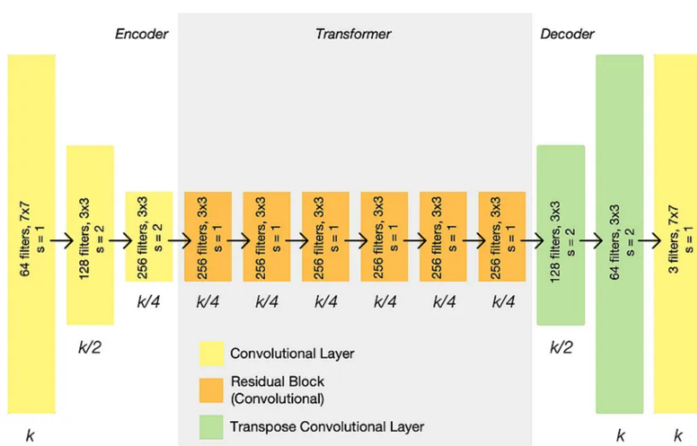

Figure 1: Generator Architecture

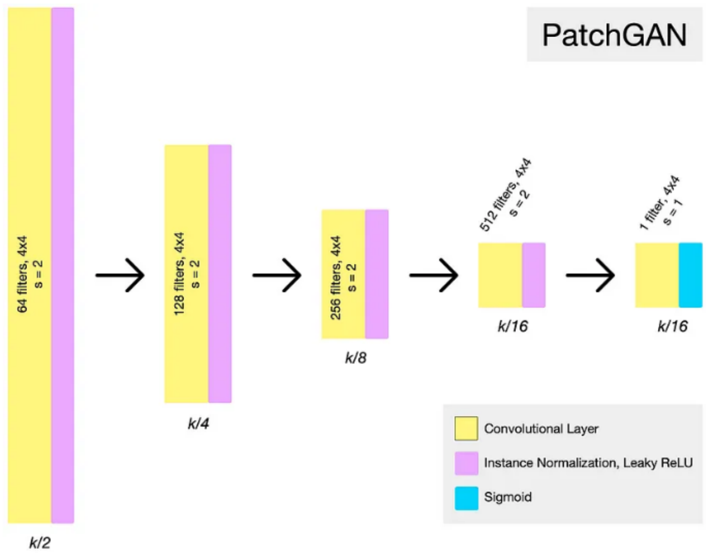

Figure 2: Discriminator Architecture

1.5 Training and Inference

The CycleGAN used for the purpose of this report is a vanilla Pytorch model. The Adam optimizer was used for the discriminator and generator of the GAN. The learning rate was set to 0.0002. Beta1 and Beta2 hyperparameters for the Adam optimizer was set to 0.5 and 0.999 respectively. The weight of the Cycle Consistency Loss was set to $\lambda = 10$. The model was trained for 10 epochs on a Nvidia T4 GPU.

During training, after every 100 steps, visual pictures that are generated are stored. Figure 1 represents generated CT scans and Figure 2 represents actual CT scans. Figure 3 represents generated MRI scans and Figure 4 represents actual MRI scans. It can be observed through comparison that the generated scans are closely similar to the actual scans. This proves the effectiveness of the Cycle-GAN in translating CT scans to MRI scans and vice versa.

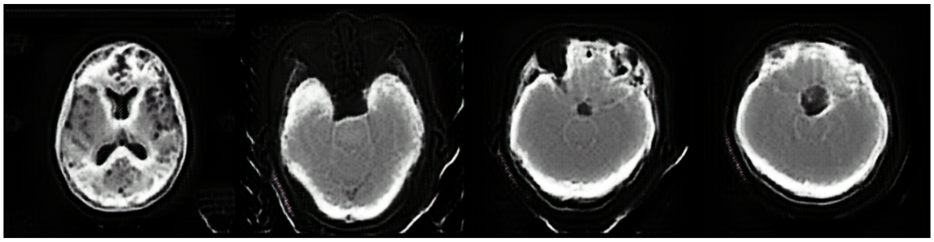

Figure 3: Generated CT Scans

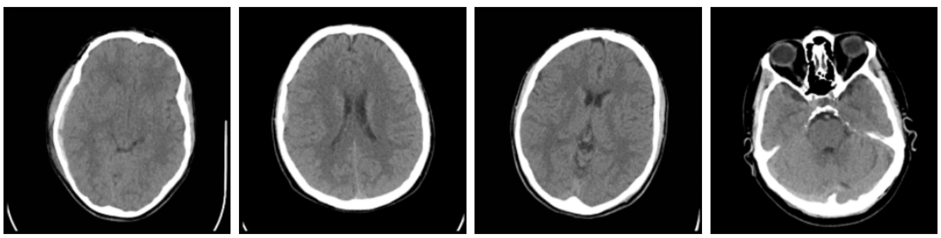

Figure 4: Random Actual CT Scans

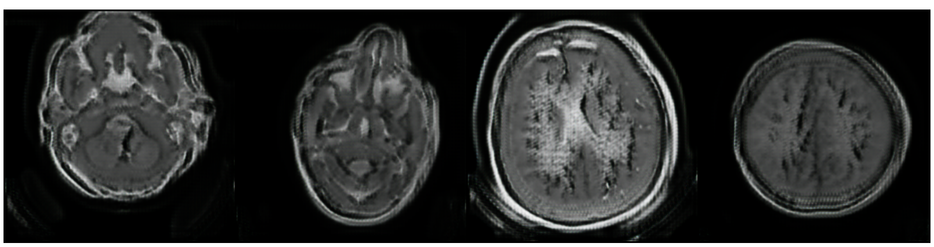

Figure 5: Generated MRI Scans

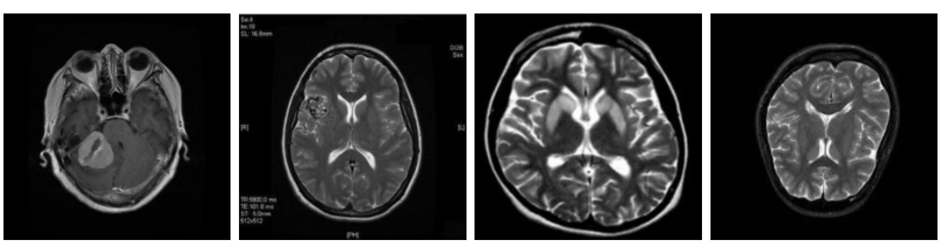

Figure 6: Random Actual MRI Scans

2 Conclusions

CT and MRI scans collected from Kaggle were successfully converted to the other using a CycleGAN. This technology can potentially prevent patients from being exposed to ionizing radiation from CT scans [1]. Qualitative results show that the model performs to a considerable degree. Further steps would be to collect quantitative results and train the model for 50-100 epochs for better accuracy. Furthermore, optimization frameworks such as ONNX Runtime can be used to significantly increase the speed of the model. The code from this paper can be found here.

References

[1] Yingchao Cai et al. "CycleGAN-based image translation from MRI to CT scans". In: *Journal of Physics: Conference Series* 2646.1 (Dec. 2023), p. 012016. DOI: 10.1088/1742-6596/2646/1/012016. URL: https://dx.doi.org/10.1088/1742-6596/2646/1/012016.

[2] Denaya. *Cyclegan-introduction + pytorch implementation*. May 2023. URL: https://medium.com/@chilldenaya/cyclegan-introduction-pytorch-implementation-5b53913741ca.