ENEL 645 FINAL PROJECT
HYPERPARAMETER OPTIMIZATION FOR CYCLEGAN

Mevin Moncy, Brandon Lac, Mustayeen Abedin, Nur-Alhuda Ali

University of Calgary, Winter 2024, Dr. Roberto Souza

ABSTRACT

The rise of generative adversarial networks (GANs) has pushed the boundaries of computer vision, that has allowed the emulation of artistic styles with remarkable fidelity. It's opened up doors to recreating artistic styles with incredible precision. Inspired by renowned artists like Claude Monet, this research endeavors to explore the potential of GANs in replicating the distinctive "je ne sais quoi" of Monet's works. The findings contribute to understanding how machine learning can emulate artistic expression, with implications for computer vision and generative modeling.

Github repository: https://github.com/MevinMoncy/ENEL-645-Final-Project

***Index Terms* —** Generative Adversarial Network, CycleGAN, Monet, Generator, Discriminator

1. INTRODUCTION

The field of computer vision has been transformed by Generative Adversarial Networks (GANs), which can create lifelike images from random noise. GANs have found applications in diverse areas, including generating photorealistic images for entertainment and advertising, producing synthetic datasets for training machine learning models, creating realistic textures in video game development, aiding in drug discovery through generating molecular structures, and developing deepfake videos for various purposes.

One notable GAN architecture, CycleGAN (Cycle-Consistent Generative Adversarial Network), has gained prominence for its capability in unpaired image-to-image translation, enabling the conversion of images from one domain to another without the need for paired training data. This feature has significant implications for applications like style transfer, photo enhancement, and domain adaptation.

The primary objective of this study is to investigate the effects of hyperparameters on the performance of CycleGAN in image-to-image translation tasks. Specifically, the study aims to explore the impact of hyperparameters such as the learning rate, lambda (λ), and identity loss coefficient on the quality of generated images, FID score, loss, and training time. By systematically varying these hyperparameters and analyzing their influence on CycleGAN's performance, this research seeks to identify optimal settings that enhance training stability, reduce training duration, and improve overall image quality.

By investigating the optimization of CycleGAN through hyperparameter tuning, this research aims to contribute to a deeper understanding of deep learning optimization techniques and their application in image-to-image translation. The findings of this study are expected to provide valuable insights into the optimization of GANs, with potential implications for enhancing image translation quality in various applications such as art generation, medical image analysis, and autonomous driving.

2. RELATED WORK

Generative Adversarial Networks (GANs) have been a subject of extensive research since their introduction by Goodfellow et al. [1] in 2014. The ability of GANs to generate realistic images has led to their widespread adoption in various domains, including computer vision, natural language processing, and beyond.

One of the key areas of focus has been the application of GANs in image-to-image translation tasks. Isola et al. [2] introduced the concept of conditional GANs for this purpose, paving the way for architectures like Pix2Pix, which rely on paired training data. However, the need for paired data limits the applicability of these models in real-world scenarios where such data is scarce.

CycleGAN, introduced by Zhu et al [3], addressed this limitation by enabling unpaired image-to-image translation. This architecture has been successfully applied in numerous applications, such as style transfer, where an image is transformed to mimic the style of a reference image, and photo enhancement, where images are enhanced to improve their quality or adjust to a specific aesthetic.

Despite its success, CycleGAN faces challenges related to mode collapse, where the generator produces limited output variations, and potential issues in preserving the core content of the original image. This study builds on the existing body of knowledge by exploring hyperparameter optimization specifically for CycleGAN, with the aim of enhancing its performance in image-to-image translation tasks.

3. MATERIALS AND METHODS

3.1 Kaggle Dataset

This study leverages the Kaggle competition dataset "I'm Something of a Painter Myself," which includes real-world photographs and corresponding Monet-style paintings [4]. The goal is to develop GAN models that can convincingly transform photographs into the artistic style of Monet. Models in the competition are evaluated using the MiFID (Memorization-informed Fréchet Inception Distance) metric. MiFID combines a measure of visual similarity to real Monet works (FID) with a penalty for simply memorizing the training images. For our study however, we will test the effectiveness of the CycleGAN model using FID. This metric compares the distribution of generated images with the distribution of a set of real images. The following table highlights the details of our dataset.

Table 1**.** Dataset Details

Detail	Value	Size
Number of Monet Images	300	256x256 jpeg
Number of photo images	7038	256x256 jpeg
Total Number of images	7338	256x256 jpeg

The dataset comprises two main categories:

- **Photo Images:** These are real-world photographs covering a wide range of scenes, including landscapes, buildings, and natural environments. They serve as the input to the GAN model for style transfer.
- **Monet Images:** These are digitized versions of Claude Monet's paintings, providing the target style that the GAN model aims to learn and replicate on the input photo images.

3.2 Data Visualization:

In this subsection, we present two figures: one showcasing a random assortment of photo and Monet images, and another demonstrating the edge detection applied to a random assortment of images.

The first figure displays a selection of real-world photographs alongside Monet-style paintings from the dataset. This visualization helps to illustrate the differences in style and content between the two categories of images, which the GAN model aims to bridge through style transfer.

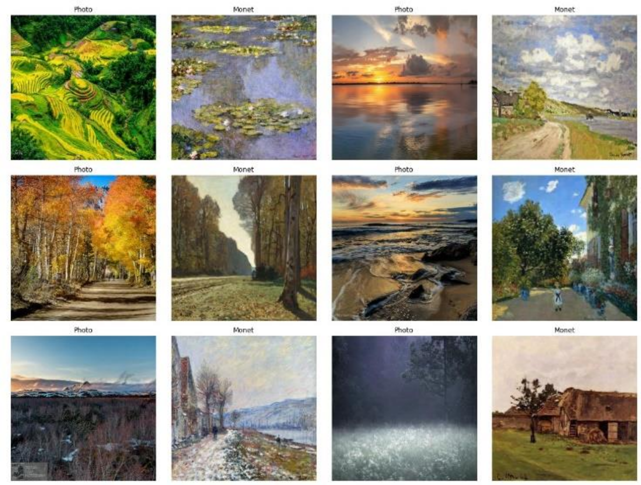

Fig. 1. Example Images from Dataset

The second figure applies edge detection to a random assortment of images from both categories. Edge detection highlights the contours and outlines within the images, providing insights into the structural and stylistic elements that the GAN model needs to capture and replicate.

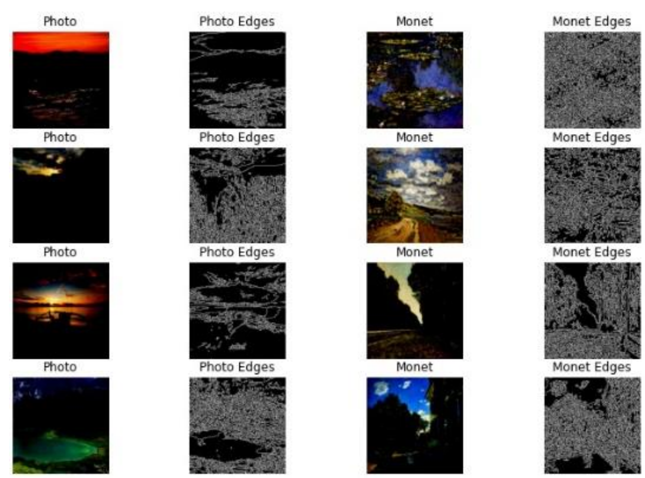

Fig. 2. Edge Detection on Sample Images

These visualizations not only aid in understanding the dataset but also in evaluating the performance of the GAN model. By comparing the edge detection results between the original and generated images, we can assess how well the model is able to preserve the essential features and structures during the style transfer process.

3.3 Data Preprocessing

Data preprocessing is a crucial step in preparing the dataset for training the Generative Adversarial Network (GAN) model. For the style transfer task using the "Getting Started with GANs" dataset from Kaggle, the following preprocessing steps were implemented:

1. **Image Resizing:** All images, both Monet paintings and real-world photographs, were resized to a uniform dimension of 256x256 pixels. This standardization is necessary to ensure that the images are compatible with the neural network architecture and to reduce computational complexity.

2. **Normalization:** The pixel values of the images were normalized to a range of [-1, 1]. This normalization helps in stabilizing the training process and is a common practice when working with GANs. The normalization formula used is `(pixel_value - 0.5) / 0.5`, where the pixel values are initially in the range [0, 1].

3. **Data Augmentation:** Data augmentation for style transferring tasks should be done carefully since it is important to keep the style of the dataset. If there are too many changes to the style (brightness, contrast, saturation), it can cause the generator to incorrectly learn the base style, so it is important to only implement spatial transformations.

3.4 Model Selection

CycleGAN is an architecture of a type of generative adversarial model designed for unpaired image-to-image translation tasks [5]. For this study, we will be using the CycleGAN implementation produced in the original paper by Zhu et al [3].

3.4.1 Generator

The primary purpose of the Generator Model is to create new data samples that are similar to those in the training data it was provided. Generators aim to produce realistic translations between domains, using upsampling and residual blocks to preserve important image characteristics, while striving to deceive the discriminators [6].

In the context of CycleGAN, there are two generators: one that takes photos and generates Monet paintings, and another that takes Monet paintings and generates photos. The Generator Model operates by learning to map random noise or latent representations to the data space, adjusting its parameters such that the generated samples become increasingly similar to the real data samples.

Upsampling is crucial for enhancing the resolution of images. Given that Monet paintings and photographs often differ significantly in terms of resolution and detail, upsampling ensures that these nuances are preserved and accurately captured during the translation process. As a result, the model can generate output images that closely mirror the target domain, be it the distinctive style of Monet paintings or the realism of photographs.

Residual blocks are essential for enabling the network to learn complex mappings between these diverse domains. Monet paintings and photographs exhibit unique characteristics in terms of details, colors, and textures. Residual blocks facilitate the effective learning of these mappings by allowing the network to concentrate on the differences or residuals between the input and target images. This approach leads to the generation of translations that are not only more accurate but also more visually compelling, bridging the gap between the two distinct artistic realms.

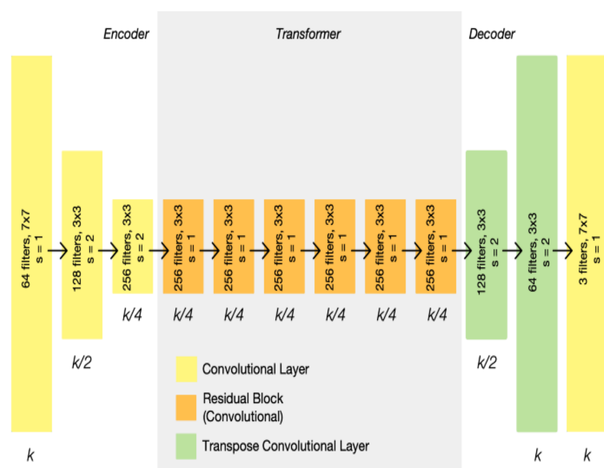

Fig. 3. Generator Model

3.4.2 Discriminator

The discriminator serves to differentiate between real data samples and artificially generated samples produced by the Generator. In CycleGAN, there are two discriminators, each specialized in evaluating the realism of images from one of the two domains involved in the translation process (e.g., Monet paintings and photographs). The discriminators are trained to distinguish between real images from their respective domains and fake images generated by the corresponding generators [7].

During the training process, the discriminators receive both real and generated images as input and output a probability score indicating the likelihood that the input image is real. This score is used to calculate the loss for both the discriminator and the generator, creating an adversarial relationship that drives the improvement of both models. The discriminators' ability to accurately assess the realism of generated images is crucial for guiding the generators to produce more realistic and convincing translations between the two domains.

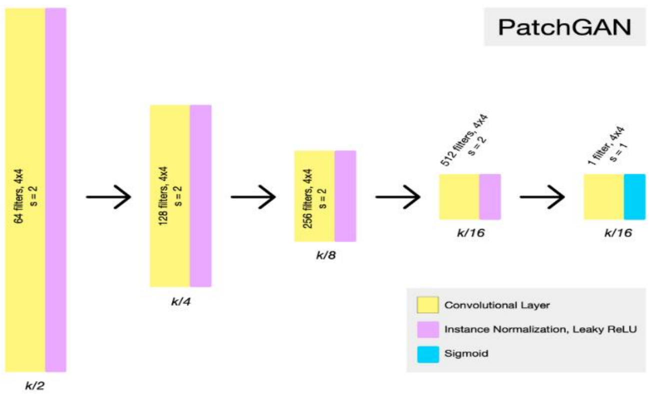

Fig. 4: Discriminator Model

3.5 Model Training

To test the effects of the key hyperparameters, seven experiments were conducted including an initial baseline model that uses recommended hyperparameters. The hyperparameters adjusted in each iteration of the experiment include the learning rate, lambda (λ), and the identity coefficient. The following table highlights the different values for each hyperparameter.

Table 2. Experiment by Hyperparameter Values

Experiment	Learning Rate	Lambda	Identity Coef
1 (baseline)	2e-4	10	0.5
2	5e-4	10	0.5
3	1e-4	10	0.5
4	2e-4	5	0.5
5	2e-4	20	0.5
6	2e-4	10	0.25
7	2e-4	10	0.75

The effects of the experiments that were tested were the total training time, the discriminator and generator loss, qualitative image quality, and the FID score. Each model was trained with a Kaggle notebook using PyTorch on their cluster which utilized their GPU. All other important hyperparameters were kept the same including a batch size of 1 and the number of epochs which was kept at fifty.

4. RESULTS AND DISCUSSION

The following section highlights our results from each model training and our analysis on the results. The table below highlights the quantitative effects for each model:

Table 3. Experiment Results

Experiment	Training Time (s)	Generator Loss	Discriminator Loss	FID
1	8224	5.9	0.15	101.5
2	9170	6.1	0.11	97.54
3	10080	6.0	0.091	100.5
4	10243	3.9	0.012	104.9
5	8243	10	0.12	99.7
6	8229	5.51	0.11	106.4
7	10361	6.69	0.11	104.4

From the results above, we can make the following observations:

- Learning Rate: For this dataset, we found that both increasing and decreasing the learning rate improved the FID score. However, the lowest FID score occurred when the learning rate was increased to 5e-4. This suggests that a higher learning rate might have the greatest impact in generating more realistic images.
- Lambda: The results suggest that increasing the weight for cycle consistency loss improves the generation of our target images. However, the generator loss increased.
- Identity Coefficient: Both increasing and decreasing the identity coefficient from the baseline experiment resulted in less realistic transformation of the input photos.

4.1 Analysing best Hyperparameter Set

Considering of the best FID score of 97.54, experiment 2 (learning rate: 5e-4, lambda: 10, identity coefficient: 0.5) had the best results in terms of image quality of the generated Monet images. During training, the losses were compared at each epoch before settling to a loss of 6.1 and 0.11 for the generator and discriminator respectively.

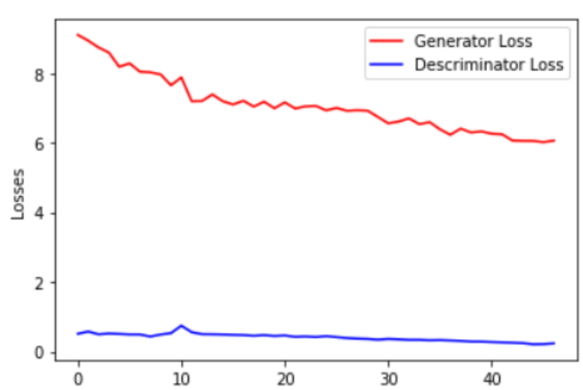

Fig 5. Losses over each Training Epoch

The following monet-esque photos were generated using the respective input photos. We observed that the form and shape of the generated images were generally kept close to the input images, but the contrast, lighting, saturation and texture were changed to resemble the Monet samples. Monet paintings have a vibrant color palette and as demonstrated by the images below, the blacks from the input photos were adjusted and nighttime images now appear closer to daytime images because of the changes to contrast, lighting and saturation. The generated images have an adjusted texture, appear softer to resemble Monet's painting style.

Fig. 6: Input compared to Generated Images

4.2 Sources of Error

The experiment was running using Kaggle Notebook on their remote clusters. There is likely to be variability in the performance due the different hardware configurations or varying loads on the system. This can lead to inconsistencies in training times as demonstrated by the wide range of training times. Things within the control of the experiment such as setting a unique seed across all experiments was used to handle the randomness of neural networks and the stochastic process of GAN training.

5. CONCLUSION

This research investigated the optimization of CycleGAN through hyperparameter tuning for the task of image-to-image translation, specifically focusing on replicating the artistic style of Claude Monet. Through tweaking key settings like the learning rate, lambda, and identity coefficient, we've explored how these changes affect the quality of the generated images and overall performance of the model. Our experiments revealed several insights into the effects of hyperparameters on CycleGAN's performance. Notably, we observed that adjusting the learning rate can significantly influence the fidelity of generated images, with a higher learning rate yielding better results in terms of FID score.

Based on our experimentation, we identified the hyperparameter configuration with the best performance, and it had an FID score of 97.54. This configuration, characterized by a higher learning rate (5e-4) and standard lambda and identity coefficient values, produced Monet-style images with enhanced visual fidelity and stylistic resemblance.

Moving forward, further research can be explored for additional hyperparameters, model architectures, and training strategies to continue improving the performance and versatility of CycleGAN and other generative models.

6. REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Nets," in *Proceedings of the International Conference on Neural Information Processing Systems*, Montreal, 2014.

[2] P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE conference on computer vision and pattern recognition*, 2017, Honolulu.

[3] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017.

[4] A. Jang, A. S. Uzsoy and P. Culliton, "I'm Something of a Painter Myself," Kaggle, 2020. [Online]. Available: https://kaggle.com/competitions/gan-getting-started.

[5] J. Brownlee, "A gentle introduction to cyclegan for image translation," 2019. [Online]. Available: https://machinelearningmastery.com/what-is-cyclegan/.

[6] A. Biswas, "What are Generative Adversarial Networks," 2023. [Online]. Available: https://www.simplilearn.com/tutorials/deep-learning-tutorial/generative-adversarial-networks-gans#what_is_a_generator.

[7] Lark, "Discriminator in GAN," 2023. [Online]. Available: https://www.larksuite.com/en_us/topics/ai-glossary/discriminator-in-gan.