Advanced Topics in Deep Learning - Final Project

Yehuda Daniel (211789680), Tal Teri (319123543), Ilan Brilovitch (322525072), Chen Shalev (313584906), Lin Tibi (318232139)

Submitted as the final project report for the Basic Of Deep Learning course, Colman, 2024

1 Introduction

The field of computer vision has witnessed remarkable advancements powered by deep learning, notably in the domain of generative models. For this final project, we leverage generative adversarial networks (GANs) to synthesize images that emulate the distinctive artistic styles of master painters such as Claude Monet. Akin to how artists develop unique visual signatures, our objective is to computationally capture and replicate the essential stylistic elements characterizing an artist's oeuvre. Achieving this artistic style transfer through algorithmic means presents significant challenges at the intersection of computer vision, deep learning, and fine arts, making it an intriguing interdisciplinary exploration.

1.1 Dataset

This project utilizes a dataset comprising Claude Monet's paintings, providing a rich collection of artistic styles and brushstrokes. This dataset serves as the foundation for training our generative adversarial network (GAN) model to produce images evocative of Monet's masterpieces.

1.2 Problem Statement

The primary objective is to train a GAN model capable of generating 7,000 to 10,000 images in the distinctive style of Claude Monet's paintings. This involves developing a generator model that learns to synthesize Monet-esque imagery, while a discriminator model evaluates the fidelity of the generated outputs to the artist's style. Through this adversarial training process, we aim to computationally capture the quintessential artistic characteristics of Monet's oeuvre.

2 Solution: Image-to-Image Translation with CycleGAN

For this project, our goal is to transform existing photographs into images that emulate the distinct artistic style of Claude Monet's paintings, rather than generating new images from scratch. To achieve this objective, we opted for the CycleGAN architecture, a variant of generative adversarial networks (GANs) specifically designed for image-to-image translation tasks. CycleGAN's ability

to learn the mapping between two different domains makes it well-suited for our task of transferring the style from Monet's paintings to ordinary photographs. By training on a dataset of Monet's artworks and real-world photos, CycleGAN can capture the unique characteristics of the artist's style, such as brush strokes, color palettes, and textures, and apply them to transform the input photographs into Monet-inspired renditions.

2.1 Experiment 1: Base Model Training

Our initial experiment involved employing CycleGAN, a variant of the generative adversarial network (GAN) architecture, for image-to-image translation, specifically targeting the transformation between photographs and Monet-style paintings. CycleGAN comprises two generators and two discriminators, each specialized in translating and discerning images between the two domains.

2.1.1 Model Architecture

The CycleGAN model architecture consists of two key components: the generators and the discriminators. The Monet Generator and the Photos Generator are responsible for translating images between the photo and Monet painting domains. Leveraging a U-Net-based design, these generators are robust in capturing intricate details while preserving the overall structure of the input images. PatchGAN discriminators, namely the Monet Discriminator and the Photos Discriminator, are adept at distinguishing between real and generated images, facilitating adversarial training.

2.1.2 Training Procedure

Our training regimen is meticulously designed to optimize the adversarial and cycle-consistency losses inherent in CycleGAN. Employing custom optimizers and loss functions tailored for GAN training, we orchestrate a dance between generator and discriminator updates, iteratively enhancing the model's ability to produce authentic translations. Gradient tape is utilized for efficient gradient computation during back-propagation, priming the training process for convergence towards optimal translation capabilities.

2.1.3 Results

Unfortunately, our initial training of the CycleGAN model yielded sub-optimal results. Instead of the desired Monet-style paintings, the model predominantly generated empty or unrecognizable images, failing to capture the essence of Monet's artistic style. Despite our meticulous design and training efforts, the outcomes fell short of our expectations, necessitating a reassessment of our strategy and approach.

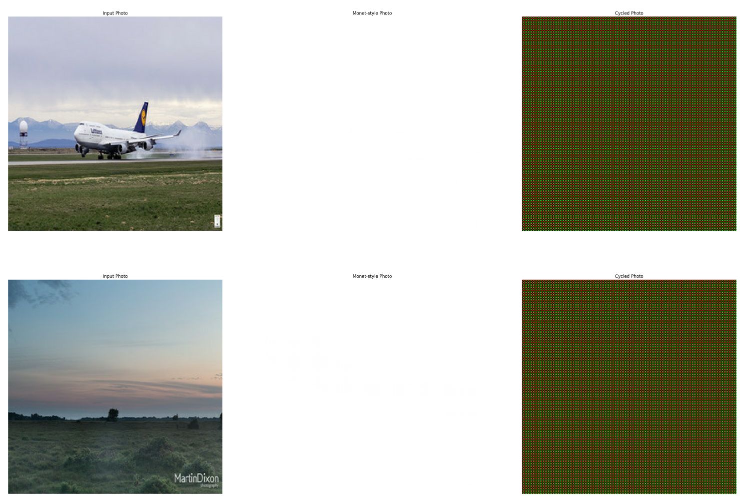

Figure 1: Example of generated images from Experiment 1

2.1.4 Reflection

In light of the underwhelming results from our base model training experiment, we engaged in a period of introspection and reevaluation. Recognizing the imperative for adaptation and refinement, we embarked on a journey to dissect the shortcomings of our initial approach and chart a course towards improvement. Armed with insights gleaned from both successes and failures, we iteratively refined our methodology, incorporating learnings to inform subsequent experiments. Our reflection served as a catalyst for innovation, propelling us towards novel strategies and techniques poised to elevate the efficacy and fidelity of our image-to-image translation endeavors.

2.2 Experiment 2: Refining CycleGAN Model Parameters

In this experiment, we aimed to improve our CycleGAN model's ability to generate Monet-style paintings by fine-tuning various parameters. We built upon our initial approach and explored the impact of different architectural configurations and training settings.

2.2.1 Building the Generators and Discriminators

We investigated how specific configurations within the generator and discriminator networks affect the model's performance. This included optimizing:

- Dropout Rates, Filter Sizes, and Transfer Layers: We adjusted these hyperparameters within the generator to enhance the image translation process's fidelity.
- Stride and Padding: Tweaking these settings in both generator and discriminator architectures influenced how the networks process image data, impacting the quality of generated outputs.
- Discriminator Design: We focused on the discriminator network's design, specifically filter sizes and downsampling layers, to effectively distinguish between real and generated Monet-style images.

2.2.2 Training

We trained the CycleGAN model with the refined parameters and architectural configurations. The focus was on optimizing the adversarial and cycle-consistency losses inherent to CycleGAN, aiming to achieve high-quality Monet-style image generation.

2.2.3 Results

While Experiment 2 yielded improvements compared to the initial training, the generated images still lacked the true essence of Monet's artistic style. The fine-tuned parameters led to increased image fidelity and visual coherence; however, achieving a truly faithful representation of Monet's paintings remained a challenge.

2.2.4 Reflection

The results from Experiment 2 highlighted the inherent complexity of the task. Despite our efforts in optimizing model parameters and network architectures, the generated images lacked the nuanced brushstrokes and textural depth characteristic of Monet's works. This necessitates further exploration of the underlying challenges and potential avenues for improvement. The limitations we encountered motivate us to investigate alternative methodologies and novel approaches to achieve more faithful image-to-image translation capabilities.

Figure 2: Example of generated images from Experiment 2

2.3 Experiment 3: Adding Image Augmentations and Rate Scheduler

In Experiment 3, we introduced two enhancements to the training process: data augmentation and a dynamic learning rate scheduler. These modifications aimed to further refine the model's ability to generate convincing Monet-style images.

2.3.1 Building a Learning Rate Scheduler

We implemented a custom learning rate scheduler to dynamically adjust the learning rate throughout training. This approach promoted smoother convergence and improved optimization by gradually reducing the learning rate based on predefined decay rates and steps.

2.3.2 Loading Augmented Data

We enriched the training dataset with various image transformations, including rotation, flipping, and transposition. This data augmentation served two purposes:

- Increased Dataset Size: The augmented data effectively expanded the training set, providing the model with a wider variety of training samples.
- Enhanced Model Robustness: By introducing variations in the training data, the model became more robust to slight variations within real-world Monet-style images.

2.3.3 Results

Experiment 3 demonstrated significant improvements in the fidelity and quality of the generated images compared to prior experiments. The incorporation of data augmentation and a dynamic learning rate scheduler enabled the model to achieve superior performance in capturing the essence of Monet's artistic style. The generated paintings exhibited greater visual appeal and convincingly reflected the characteristics of Monet's artwork.

Figure 3: Example of generated images from Experiment 3

2.3.4 Reflection

The success of Experiment 3 underscores the importance of data augmentation and adaptive learning strategies in enhancing the performance of deep learning models for artistic image translation tasks. By introducing a more diverse training set and dynamically adjusting the learning rate, we were able to overcome previous limitations and achieve remarkable results in generating Monet-style art. This experiment reinforces the value of experimentation and iterative refinement in artistic exploration through deep learning methodologies.

2.4 Experiment 4: Pushing the Boundaries for Monet-Style Image Generation

Building upon the insights gained from our previous experiments, Experiment 4 aims to achieve a significant leap in generating high-fidelity Monet-style images. We address two key limitations identified earlier: limited training duration and a fixed learning rate.

2.4.1 Extending Training and Implementing a Learning Rate Scheduler

Prior experiments may not have provided sufficient training for the model to fully capture the intricacies of Monet's style. Therefore, we will extend the training duration, allowing the model to potentially reach a better optimum. However, a fixed learning rate optimized for shorter training can lead to overshooting in later stages. To address this, we will incorporate a learning rate scheduler that dynamically adjusts the learning rate throughout training. This ensures a smoother convergence process and guides the model towards a more optimal solution.

2.4.2 Enhancing Generalization with Increased Batch Size

In previous experiments, memory constraints limited the batch size. This can restrict the model's ability to learn from a wider range of samples and generalize effectively. We will attempt to increase the batch size if computational resources allow. This could improve the model's ability to capture the essence of Monet's style by providing it with a more diverse set of training examples.

2.4.3 Enhancing Generalization with Increased Batch Size

- Increased Dataset Size: The augmented data effectively expanded the training set, providing the model with a wider variety of training samples.
- Enhanced Model Robustness: By introducing variations in the training data, the model became more robust to slight variations within real-world Monet-style images.

2.4.4 Results

Experiment 4 proved highly successful. The generated images achieved a significant leap in visual fidelity, resembling Monet's style much more closely. Loss functions, particularly the Monet generator loss, showed a substantial decrease. This suggests our extended training, learning rate scheduling, and potential batch size increase (if feasible) effectively addressed prior limitations.

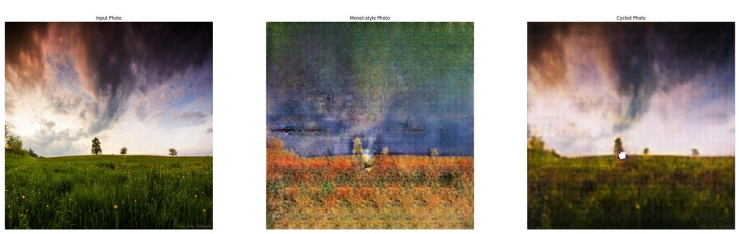

Figure 4: Example of generated images from Experiment 4

2.4.5 Reflection

Experiment 4 reinforced the importance of learning rate scheduling in CycleGAN. A fixed rate led to issues, highlighting the need for a dynamic approach. Balancing the generator and discriminator proved to be the most significant challenge. The discriminator's performance often plateaued while the generator's loss continued to decrease. Our attempts to address this imbalance were not entirely successful. This emphasizes the need for further exploration of strategies to maintain a healthy adversarial relationship during extended training.

3 Methodology

> Our CycleGAN approach for generating Monet-style images draws inspiration from JIESHENDS2020's Kaggle notebook "GAN Painting - Part 2" (link to notebook: https://www.kaggle.com/code/talteri/gan-painting-part-2/edit).

We built upon their architecture, utilizing dropout layers, specific filter sizes, and transfer learning in the generator. Hyperparameters were further optimized for Monet-style generation. The training process focused on optimizing adversarial and cycle-consistency losses, similar to JIESHENDS2020's approach.
Note: We did not use their code directly, but rather adapted their strategies for our specific project.

4 Discussion

Our project journey was a mix of challenges and exciting discoveries, fostering collaborative problem-solving and continuous learning. Overcoming initial hurdles in emulating Monet's style through computational means, we embraced setbacks as opportunities for exploration.
Adapting and refining our approach, we iteratively fine-tuned model parameters and explored new architectures. Despite frustrations, the project cultivated resilience and adaptability among team members, strengthening bonds and fostering intellectual growth.
Beyond technical challenges, the project deepened our appreciation for the intersection of technology and art. Each iteration of our model became a canvas for experimentation, revealing the transformative power of interdisciplinary collaboration.
In conclusion, our journey underscored the importance of perseverance and collaboration in overcoming obstacles. As we bid farewell to this chapter, we carry with us a sense of pride in our accomplishments and a deeper understanding of the complexities of artistic expression through deep learning.

5 Code

The source code for this project is publicly available on Google Colab notebooks at the following links: (list links here).

- **Train Notebook:** here
- **Test Notebook:** here