I'm Something of a Painter Myself

Use GANs to create art - will you be the next Monet?

Cristina Zhang

https://colab.research.google.com/drive/136n-SzH8Y7yRaAljF1Wh8qYI3N0d-Ucd?usp=sharing

Abstract

CycleGAN is a state-of-the-art framework for unpaired image-to-image translation, demonstrating success in applications like style transfer. This report investigates the use of CycleGAN for translating photos to Monet-style artwork and vice versa. Performance was analyzed by training the model for 25 epochs, focusing on generator and discriminator loss convergence. Visual results and training dynamics are presented, alongside a discussion of stability and limitations.

1 Introduction

Image-to-image translation is a machine learning task in which a model learns to map images from one domain to another. CycleGAN, proposed by Zhu et al. [2], eliminates the need for paired training data by introducing cycle-consistency loss. This report evaluates the performance of CycleGAN in transforming photos into Monet-style paintings and vice versa. The generator architecture leverages downsampling and upsampling layers for feature extraction and reconstruction, while the discriminator assesses real versus generated samples. This work analyzes the translation quality and training stability when applied to photo-to-artwork generation.

2 Methods

2.1 Data Preprocessing

The dataset used consists of two domains: photos and Monet-style paintings. The preprocessing steps included:

1. **Data Collection:** A curated dataset of images sourced from Kaggle.
2. **Loading Data:** The datasets were loaded from `monet_jpg` and `photo_jpg` directories.
3. **Resizing:** All images were resized to 256×256 pixels to ensure consistency across the training set.
4. **Rescaling:** Image pixel values were normalized to the range $[-1, 1]$ using the formula:

$$\text{Normalized Pixel Value} = \frac{\text{Original Pixel Value}}{127.5} - 1 \tag{1}$$

   This normalization helps stabilize the training process.

2.2 Model Architecture

2.2.1 Generator

The generator is based on a U-Net-inspired architecture with ResNet Blocks and skip connections. Its design ensures efficient feature extraction, preservation of spatial details, and effective image translation. It consists of:

- **Downsampling:** A series of convolutional layers with stride 2 to reduce spatial resolution while increasing feature depth.
- **ResNet Blocks:** Residual blocks are integrated to enhance the generator's ability to learn complex features while mitigating the risk of vanishing gradients. Each ResNet Block consists of:
  - Two convolutional layers with a kernel size of 3×3.
  - Batch normalization after each convolutional layer.
  - A residual connection that adds the input to the output of the block.

- **Upsampling:** Transposed convolutional layers with stride 2 to reconstruct the original resolution.
- **Skip Connections:** Features from the downsampling layers are concatenated with corresponding upsampling layers to retain spatial details and improve reconstruction accuracy.

The generator outputs an image in the target domain that mimics the style and characteristics of the target domain while preserving the content structure from the input image[1].

2.2.2 Discriminator

The discriminator uses a PatchGAN architecture, which classifies 70×70 patches as real or fake instead of the entire image. This allows the model to focus on local features and improves its ability to detect fine-grained artifacts.

2.3 Loss Functions

Figure 1 illustrates the schematic diagram of CycleGAN principles. The domains X and Y represent the data sets of realistic landscape images and art paintings, respectively. D_X and D_Y are discriminators tasked with distinguishing real images from generated ones in their respective domains. G_X and G_Y are generators that perform the translations between these two domains.

The loss functions used in CycleGAN are inspired by the framework proposed by Jun-Yan Zhu et al. [2], aiming to achieve effective style transfer through adversarial training and cycle consistency.

Cycle-Consistency Loss The cycle-consistency loss ensures that the original input is preserved after a cycle of translations through both generators. It is defined as:

$$\mathcal{L}_{cc} = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[||G_Y(G_X(x)) - x|| \right] + \mathbb{E}_{y \sim p_{\text{data}}(y)} \left[||G_X(G_Y(y)) - y|| \right]. \tag{2}$$

This loss penalizes differences between the input and the reconstructed output, preserving the content during style transformation.

Adversarial Loss The adversarial loss drives the generators to produce outputs that the discriminators cannot distinguish from real images. For the generator G_X

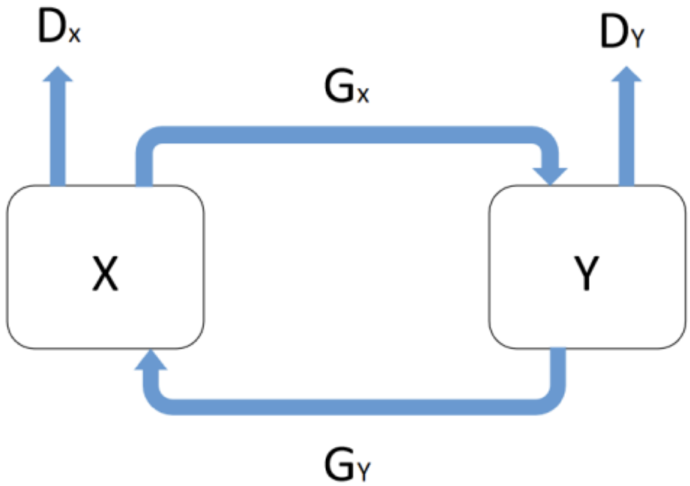

Figure 1: Schematic diagram of CycleGAN principles.

and discriminator D_Y, the adversarial loss is:

$$\mathcal{L}_{adv}(G_X, D_Y, X, Y) = \mathbb{E}_{y\sim p_{\text{data}}(y)}\left[\log D_Y(y)\right] + \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log(1 - D_Y(G_X(x)))\right]. \tag{3}$$

Similarly, the adversarial loss for G_Y and D_X is:

$$\mathcal{L}_{adv}(G_Y, D_X, X, Y) = \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log D_X(x)\right] + \mathbb{E}_{y\sim p_{\text{data}}(y)}\left[\log(1 - D_X(G_Y(y)))\right]. \tag{4}$$

Identity Loss The identity loss encourages generators to preserve the content of images already in the target domain. It is defined as:

$$\mathcal{L}_{id}(G_X, G_Y) = \mathbb{E}_{y\sim p_{\text{data}}(y)}\left[||G_X(y) - y||\right] + \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[||G_Y(x) - x||\right]. \tag{5}$$

CycleGAN Loss The total loss for CycleGAN is the sum of the above components:

$$\mathcal{L}_{CycleGAN} = \mathcal{L}_{cc} + \mathcal{L}_{adv}(G_X, D_Y, X, Y) + \mathcal{L}_{adv}(G_Y, D_X, X, Y) + \mathcal{L}_{id}(G_X, G_Y). \tag{6}$$

This combined loss function ensures that the model learns to translate images effectively between domains while maintaining the consistency and identity of the input data[1][2].

2.4 Training

The model was trained for 25 epochs using the Adam optimizer with a learning rate of 2×10^{-4}. The training dataset was processed in batches of size 1. During training, discriminator losses ($\mathcal{L}D_X$ and $\mathcal{L}D_Y$) and generator losses ($\mathcal{L}G$ and $\mathcal{L}F$) were monitored for convergence.

3 Results

3.1 Visual Results

The CycleGAN model was trained to perform bidirectional image-to-image translation between two domains: **Monet-style paintings** and **photos**. The results highlight the model's ability to generate convincing transformations while also revealing some challenges and limitations.

3.1.1 Photo-to-Monet Transformation

The Photo-to-Monet transformation uses generator G_X to translate realistic photos into Monet-style artwork. Figure 2 illustrates the generated outputs. The model mimics Monet's artistic style, characterized by:

- Color palettes with warm and soft tones.
- Smooth extraction and transitions that replicate Monet's techniques.
- Preservation of the overall composition and scene structure of the original image.

However, in some instances, the transformations lack sufficient abstraction or fail to adapt fine details effectively. For example, highly textured regions such as tree canopies or water reflections may show minimal stylistic changes, leading to results that appear overly similar to the original photo.

3.1.2 Monet-to-Photo Transformation

The Monet-to-Photo transformation utilizes generator G_Y to translate Monet-style paintings into realistic photos. Figure 3 demonstrates the results. The model achieves:

- Realistic textures and details, converting brush strokes into natural patterns.
- Improved lighting and shadow rendering, giving the output a lifelike appearance.

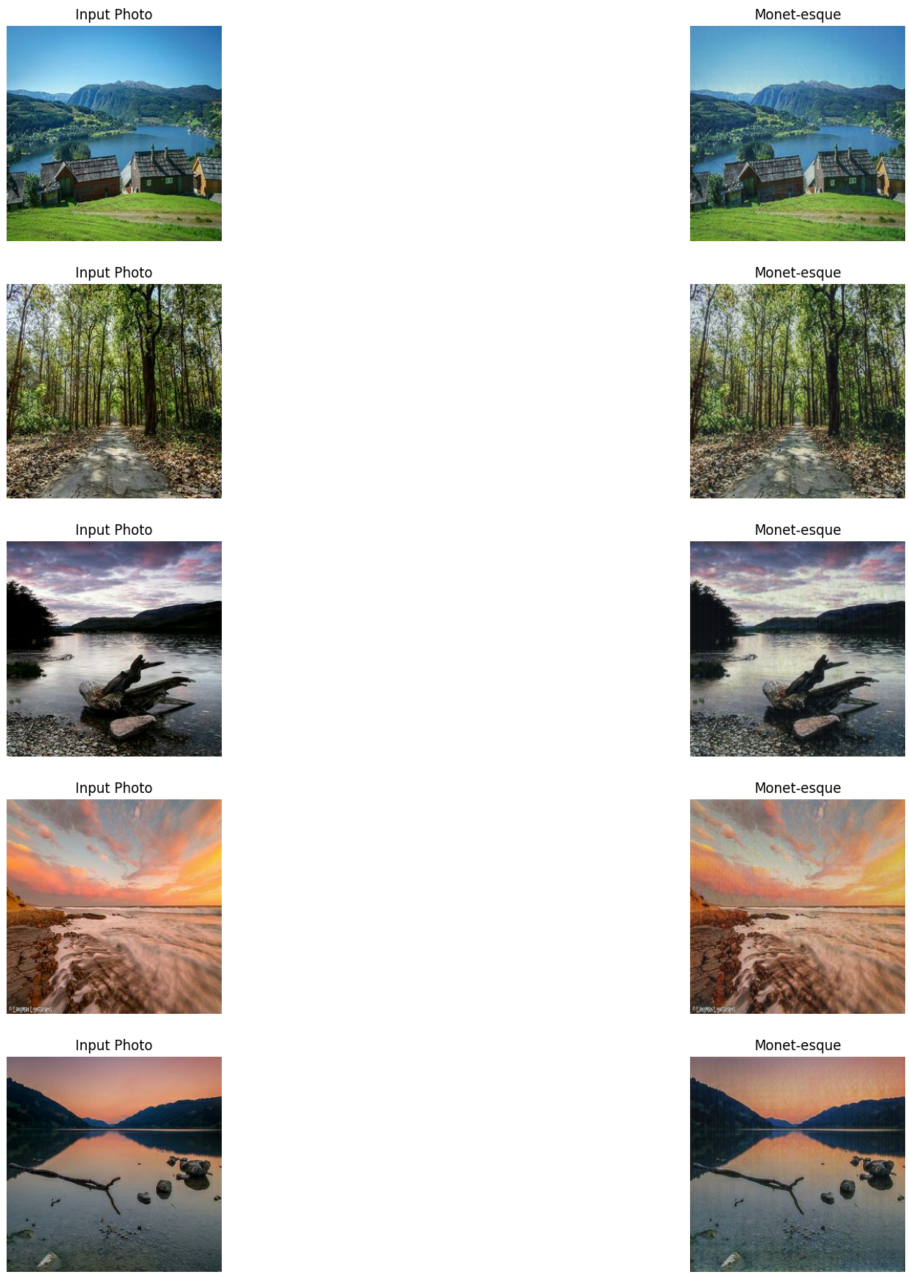

Figure 2: Photo-to-Monet translation examples. Left: Input photos. Right: Monet-style outputs.

- Consistency in retaining the composition and major elements of the original paintings.

While most transformations effectively convert Monet's paintings into photo-like images, certain outputs exhibit minimal differences, particularly in high-abstraction regions such as foggy or distant landscapes. This suggests that the model struggles to fully recover fine-grained details from highly abstract artistic inputs.

3.1.3 Analysis of Observed Limitations

The visual results reveal the following limitations:

- **Low Contrast in Transformations:** In some cases, the translated outputs show only slight variations from the input, indicating that the model might be underfitting or unable to capture intricate domain-specific features.

- **Artifacts:** Certain transformations introduce visual inconsistencies, such as overly smoothed textures or distorted small details, which detract from the realism of the output.

- **Domain Gaps:** For Monet-to-Photo transformations, the model struggles with abstract features (e.g., brush strokes or foggy scenes), which leads to partial reconstruction of photo-like details.

3.2 Training Dynamics

The CycleGAN model was trained for 25 epochs, and the generator and discriminator losses were monitored throughout the training process. Figure 4 illustrates the trends of these losses over the training period for both the Monet-to-Photo and Photo-to-Monet transformations.

- **Generator Losses** represented by the blue (Monet Generator Loss) and orange (Photo Generator Loss) curves in Figure 4, exhibit a steady decline during the initial epochs. This indicates that the generators G_X and G_Y are progressively learning to produce outputs that align closely with the target domains.

- **Discriminator Losses** shown by the green (Monet Discriminator Loss) and red (Photo Discriminator Loss) curves, remain relatively low throughout the training process. This indicates that the discriminators D_X and D_Y are consistently able to distinguish real images from generated ones.

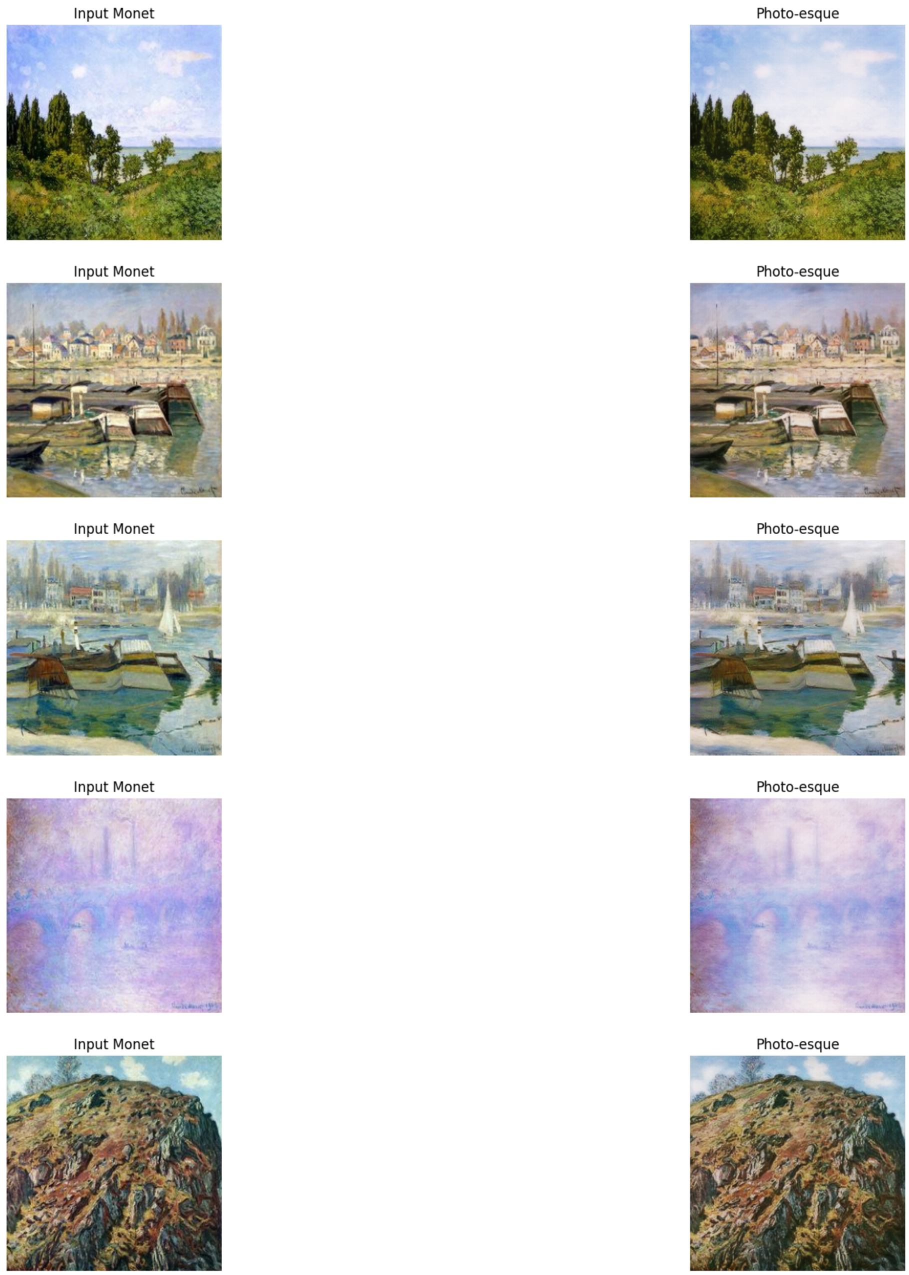

Figure 3: Monet-to-Photo translation examples. Left: Input Monet paintings. Right: Photo-like outputs.

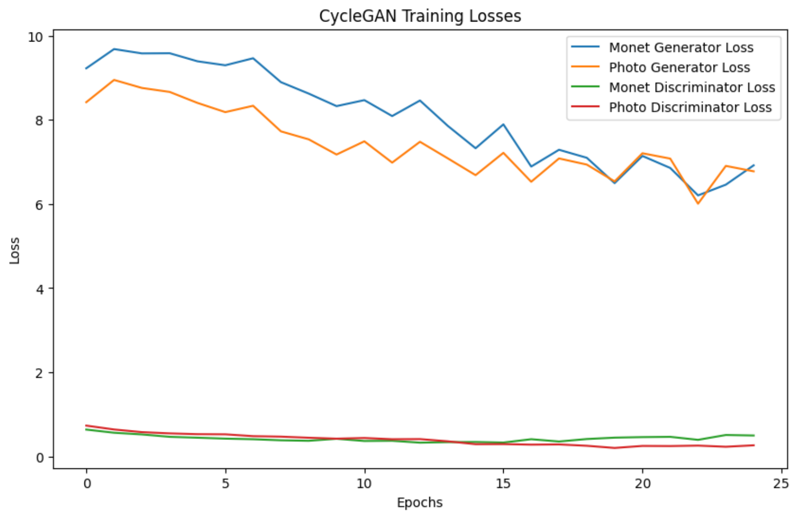

Figure 4: Training dynamics of CycleGAN. The plot shows the generator and discriminator losses for both domains over 25 epochs.

The training dynamics reveal important aspects of the model's performance:

- **Training Stability:** The consistent trends in both generator and discriminator losses reflect a well-balanced adversarial training process.
- **Fluctuations in Losses:** The minor fluctuations in the later epochs highlight the adversarial nature of GAN training, where the two networks continuously compete.
- **Scope for Improvement:** Techniques such as learning rate scheduling, weight regularization, or improved loss weighting could help mitigate fluctuations and achieve more stable convergence.

4 Discussion

The CycleGAN model demonstrated the ability to perform bidirectional image-to-image translation between photos and Monet-style paintings without paired training data. The use of generators and discriminators, coupled with cycle-consistency and identity loss functions, allowed for effective style transfer and domain adaptation.

The visual results indicate that the model successfully captures stylistic features such as Monet's brush strokes and color palette. However, in some cases, the transformations showed minimal differences, particularly in regions with high texture or abstraction, highlighting areas where the model struggles to fully align with the target domain's characteristics. Artifacts were also observed in certain outputs, suggesting room for improvement in preserving fine details.

The training dynamics reveal well-balanced adversarial learning, with consistent discriminator performance and relatively stable generator losses. Nonetheless, fluctuations in generator losses during later epochs suggest challenges in convergence, possibly due to the adversarial nature of GAN training.

Future improvements could include refining the weighting of loss functions, incorporating self-attention mechanisms for better global feature understanding, and employing adaptive learning rates to stabilize training. These enhancements have the potential to further improve the quality of transformations and address current limitations.

References

1. Zhu, J., et al. (2017). "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks." *GitHub Repository.* Available at: `https://github.com/junyanz/CycleGAN`.

2. Lux, O. "Something of a Painter: Using CycleGAN for Artistic Style Transfer." *GitHub Repository.* Available at: `https://github.com/omerlux/Something-of-a-Painter`.

3. Keras Team. "CycleGAN: Example for Generative Models with Keras." *Keras Documentation.* Available at: `https://keras.io/examples/generative/cyclegan/`.

4. TensorFlow Team. "Deep Convolutional GAN (DCGAN)." *TensorFlow Tutorials.* Available at: `https://www.tensorflow.org/tutorials/generative/dcgan`.

5. TensorFlow Team. "CycleGAN: Tutorial for Generative Models." *TensorFlow Tutorials.* Available at: `https://www.tensorflow.org/tutorials/generative/cyclegan`.

References

[1] Xu Ma. A comparison of art style transfer in cycle-gan based on different generators. In *Journal of Physics: Conference Series*, volume 2711, page 012006. IOP Publishing, 2024.

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.