Transfer Your Photo to Ghibli Cartoon Style

Shihao Piao
Department of ECE
University of Toronto
shihao.piao@mail.utoronto.ca

Yang Song
Department of ECE
University of Toronto
viola.song@mail.utoronto.ca

Yu Gu
Department of ECE
University of Toronto
raymond.gu@mail.utoronto.ca

Xuwen Shuai
Department of ECE
University of Toronto
xuwen.shuai@mail.utoronto.ca

Abstract

Our research identified a trend in the growing need for image-style processing among the younger generation. The main target of our project is to build a deep learning model with the capability of transforming an original photo into a Ghibli Cartoon style using a Generative Adversarial Network (GAN), which involves style transfer or image-to-image translation. The comprehensive architecture design includes a step-by-step approach from data processing, generator design, testing, model training, and tuning. The result is straightforward and can be visually inspected and measured by numerical metrics such as Fréchet inception distance.

Assentation of Teamwork

1. Data Collection, Implement baseline(NST) (25%) - Yang Song
2. Designing the Generator and Discriminator (25%) - Yu Gu
3. Loss and train, test code (25%) - Shihao Piao
4. Train and finetune the model, Numerical test (25%) - Xuwen Shuai

1 Introduction

Style transfer is an important area of modern research not only in digital image processing and computer vision, but also at the intersection of technology and art. Style transfer is a technique for compositing the style of an image on the content of another. It synthesises two images together so that the content and style are blended in a manner that's coherent to the eye.

1.1 Target problem:

The target problem of our project is to conduct style transfer on original images, transform them into images that emulate the Ghibli style, and evaluate our model's performance in achieving this artistic translation. This endeavor pays homage to Studio Ghibli's artistry and explores the capabilities of contemporary machine-learning techniques in capturing and replicating complex artistic styles.

1.2 Motivation:

Baseline: Neural Style Transfer (NST)

Our initial approach was to use Neural Style Transfer (NST), a popular technique that has shown remarkable success in applying the stylistic features of one image to the content of another. NST operates by optimizing a content image to reflect the style of a given style image through deep neural networks, making it a potent tool for artistic style emulation.

However, while NST provides a solid foundation for style transfer, it has limitations.The quality of NST's output is heavily contingent upon the chosen reference style image, resulting in variable model

performance. This dependence introduces an element of unpredictability and inconsistency in the stylistic output, detracting from the model's reliability for consistent style application.

Proposed Solution: CycleGAN
In response to the limitations identified with NST, we propose the implementation of Cycle-Consistent Generative Adversarial Networks (CycleGAN) as a superior alternative. CycleGAN presents a groundbreaking framework for image-to-image translation without necessitating paired examples. CycleGAN trains a model using a diverse array of style images, resulting in higher robustness, and the output of the style transfer process is independent of any single inputted style image

CycleGAN operates on the principle of cycle consistency, meaning that it can learn to translate an image from one domain (real-world landscapes) to another (Ghibli style) and back again, ensuring that the original image can be recovered. This process not only allows for the effective transfer of styles between unpaired images but also encourages the preservation of content integrity during the transformation process.

2 Problem Specification

In pursuit of our goal to conduct style transfer on real-world images, transforming them into the distinctive and whimsical style characteristic of Studio Ghibli's animations, our project unfolded in several strategic phases.

A. Data Collection and Baseline

Data Collection: We employed a web crawler to systematically gather images from the internet as our dataset for our project.
Baseline (NST): As a foundational step, we established Neural Style Transfer (NST) as our baseline model.

B. Designing the Generator and Discriminator

CycleGAN Architecture: Moving beyond the baseline, we implemented the CycleGAN model, focusing on the Generator and the Discriminator. The Generator's role is to transform images from one domain (real-world landscapes) to another (Ghibli style), and vice versa.

C. Loss Functions

Our training process incorporates three key loss functions to refine the model's performance: Adversarial Loss， Cycle Consistency Loss, and Identity Loss.

D. Model Evaluation

Numerical Evaluation: For quantitative analysis, we utilized the Fréchet Inception Distance (FID) metric to measure the similarity between generated images and real images within each domain.
Manual Evaluation: Alongside numerical testing, we conducted manual evaluations of the generated images.

3 Design Details

3.1.Data Collection and Preprocessing:

3.1.1 Data Collection

The cornerstone of our project's methodology lies in the comprehensive collection and assembly of a robust dataset, pivotal for the training and testing of our CycleGAN model. To achieve a dataset that accurately represents both authentic landscapes and Ghibli-style images, we employed a web crawler.
Web Crawler Implementation: Our data collection process was initiated through the deployment of a web crawler designed to navigate and extract images from specific online resources. This automated tool was meticulously programmed to filter and download images that specifically match our criteria for authentic landscape and Ghibli-style images. The crawler was configured to recognize and retrieve images with '.png' and '.jpg' extensions, ensuring the content's compatibility with our processing tools.The dataset merges images sourced directly from websites [4] with those obtained via the crawler. Through this method, we successfully amassed approximately 1,800 authentic landscape images for the training set

(trainX), about 900 Ghibli-style images for the training set (trainY), around 700 authentic landscape images for the testing set (testX), and roughly 30 Ghibli-style images for the testing set (testY).

3.1.2 Data Preprocessing

Once the images were collected, the next critical phase involved their preprocessing. This step is crucial to preparing the dataset for the CycleGAN model, ensuring uniformity in image dimensions and enhancing the model's ability to learn effectively.

Image Transformation: To standardize the input for the CycleGAN model, each image underwent a series of transformations, including resizing to a uniform scale, random cropping to a predetermined size, conversion into tensor format, and normalization of pixel values. These transformations were crucial for model training efficiency and performance.

3.2. Designing the Generator and Discriminator Networks:

Generator:

For the project we implemented the Generator for a Generative Adversarial Network, and the network is composed of three main parts: encoder, transformer, and decoder. Here's a breakdown overview.

1. **Encoder:** The encoder consists of convolutional layers to downsample the input, thereby reducing its spatial dimensions while increasing the depth. Each downsample step will halve the spatial dimensions (from k to k/4) and double the depth of the feature maps.
2. **Transformer:** The transformer part contains several residual blocks that maintain the same dimensionality. The residual blocks allow for the training of deep networks by using skip connections to jump over some layers. These blocks help in addressing the vanishing gradient problem by allowing the gradient to flow through the network.
3. **Decoder**: the decoder consists of transpose convolutional layers, which perform the reverse operation of the convolutional layers to upsample the feature maps back to the original input size.

Implementation of Generator:

1. **Initialization** - the initialization of the generator class takes all required parameters for initial configuration, including input/output channels, number of filters, number of residual blocks, number of downsampling layers, and whether or not dropout is applied.
2. **Model Building** - we implemented the model building for padding, instance normalization, and ReLU activation, the filter is doubled after downsampling. The last layer is also convolution reducing the number of filters to 1 with the output that can be classified for each patch of image.
3. **Forward Pass** - The forward pass simply takes an input x and runs it through the model.

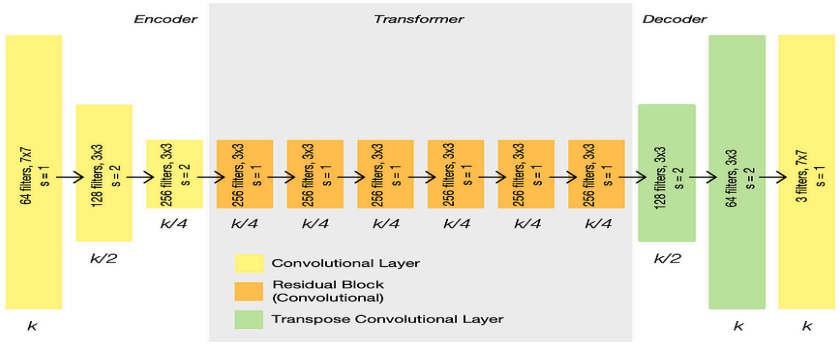


FIGURE 1. Architecture of the Generator

Fig 1 refers to the architecture enables the network to transform input images by encoding them down to a lower-dimensional space, processing them through a series of transformations, and then decoding them back to image space.

Discriminator:

For our project to perform image-to-image translation, we implemented a discriminator similar to PatchGAN that outputs a classification for each patch of the image. The discriminator classifies patches of the image to distinguish real images from generated images. This architecture doesn't classify the entire image as real or fake but instead classifies patches of the image to distinguish real images from generated images.

Implementation of Discriminator:

1. **Initialization** - the initialization of the generator class takes all required parameters for initial configuration, including the number of channels, filters and convolutional layers, kernel size, and paddings applied.
2. **Model Building** - the model building function uses a convolutional layer with LeakyReLU activation. The final output filter is 1, allowing the final output to be interpreted as the classification for each patch of the image.
3. **Forward Pass** - The forward pass function runs an input x through the model to produce the output.

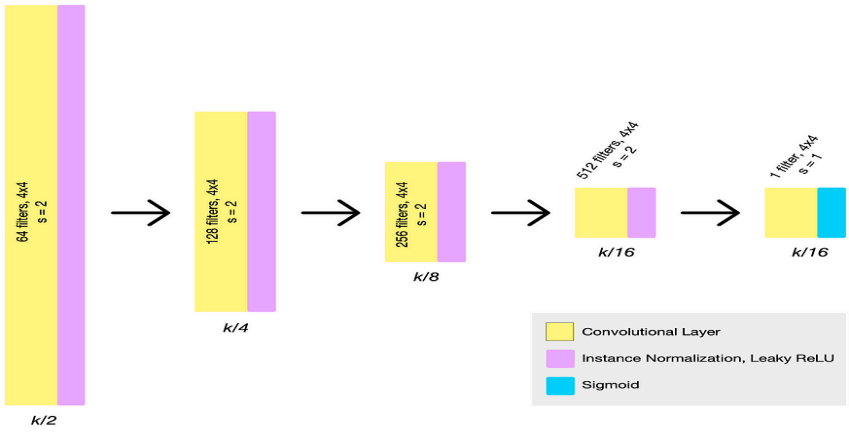


FIGURE 2. Architecture of the Discriminator

Fig.2 refers to the architecture designed to output a matrix of scores, with each element corresponding to a patch in the input image. High scores indicate that the discriminator thinks the patch is real, while low scores indicate the patch is fake.

3.3.Implementing the Loss:

For convenience, we assume real landscape image as domain A and Ghibli style image as domain B

3.3.1 Adversarial loss

The adversarial loss in CycleGANs is of the standard GAN type: here, one has a generator (G) that tries to generate images similar to the target domain and a discriminator (D) that attempts to distinguish between real images from the target domain and fake images coming from the generator. The adversarial loss for each generator is calculated as follows. The adversarial loss (A → B) is the loss of generator G with respect to its ability to fool discriminator D_B such that the generated images (G(A)) look like they belong to domain B. In practice, it is usually computed by binary cross-entropy loss, where D_B should predict 1 for real images and 0 for fake ones, but this time in our project, we chose MSE.

For Generator F (B → A): Similarly, the adversarial loss for F measures how well F can fool D_A into believing that the generated images (F(B)) belong to domain A, with a similar binary cross-entropy loss. The formula of the loss is shown in 3.1 and 3.2.

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[D(G(x))^2 \right] + \mathbb{E}_{y \sim p_{\text{data}}(y)} \left[(D(y) - 1)^2 \right] \qquad (3.1)$$

$$\mathcal{L}_{GAN}(F, D_X, Y, X) = \mathbb{E}_{y \sim p_{\text{data}}(y)} \left[D(F(y))^2 \right] + \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[(F(x) - 1)^2 \right] \qquad (3.2)$$

3.3.2 Cycle consistency loss

The cycle consistency loss allows the translated image from domain A to B and then back to A (or from B to A and then back to B) to be similar to the original image. This loss is computed based on the absolute error (L1 loss) or mean squared error (L2 loss) of the original image to the cycled image. In other words, for images A and B above, cycle consistency loss will look like this:

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\|F(G(x)) - x\|_1 \right] + \mathbb{E}_{y \sim p_{\text{data}}(y)} \left[\|G(F(y)) - y\|_1 \right] \qquad (3.3)$$

3.3.3 Identity loss

The identity loss is used to regularize the model further and helps the generator to be an identity function whenever an image from the target domain is given as input. The loss is also calculated with the L1 loss, which tries to minimize the difference between an image from the target domain and its generation by the generator of the same domain. For the generator G (B → B), the loss is calculated as the L1 loss between B and G(B), which shows that feeding B to G should produce an image close to B. Whereas for Generator F (A → A), as with the previous case, the L1 loss between A and F(A) indicates that feeding A to F results in an image resembling A.

$$\mathcal{L}_{iden}(G, F) = \mathbb{E}_{y \sim p_{\text{data}}(y)} [\|G(y) - y\|_1] + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\|F(x) - x\|_1] \tag{3.4}$$

3.3.4 Overall loss

The overall loss function is a weighted sum of these components, with hyperparameters to control the relative importance of each loss type. We chose 10 as the cycle loss weight and 0.5 as the identity loss weight.

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda_{cyc}\mathcal{L}_{cyc}(G, F) + \lambda_{iden}\mathcal{L}_{iden}(G, F) \tag{3.5}$$

3.4.Training the Model:

Model training was done on GPU on the Google Colab platform for over 30 hours. The training loop, backpropagation, and optimization process for training were implemented using PyTorch with the Adam optimizer. The network was trained for 100 epochs, with 70 using a constant learning rate and 30 using linearly decreasing learning rates. The result was systematically saved after every 10 epochs to ensure the progress was saved. With the Colab session we scheduled, we were used to Colab's interruption. If that session were interrupted, training would be resumed from the most recent saved model to minimize the loss of progress.

3.5.Another approach: Neural Style Transfer (NST):

Neural Style Transfer (NST)[2] is a technique that employs the capability of Convolutional Neural Networks (CNNs) to abstract features from two input images separately for blending image content from one image with the style representation of another. We need two images: one to transfer your style onto, which will provide the structure of the final output, and another image—the style pattern you would like to see in the final output. Another usually used pre-trained CNN (we used) would be VGG-19. This has been trained over massive datasets for image classification and has learned rich feature representations of a wide range of images. The CNN processes both the content and style images, producing a set of feature maps at each network layer. Starting from the simple features of an image, like edges and textures, the first layers up to much more complex ones, the latter constitute high-level content. This selection of a pertinent layer to represent a certain level of information in the image becomes possible after these operations. These layers are generally deeper into the network, where the high-level features are captured. We select the fourth layer, which stands as a feature for content. Then, we selected all layers that would be used for the style of the image. These features are low-level and high-level, and they extract the style by computing the Gram matrix for each layer. The Gram matrix that computed the correlation of different feature maps was applied to measure the correlation, efficiently capturing the distribution of patterns and textures. For the loss function, we have considered MSE. Compute the total loss and the weighted style loss with content loss. In each iteration, the input image is updated so that the content loss and style loss decrease with each subsequent input image. Several iterations of this process follow until the image content begins to look like the content image, while at the same time, the style of the image starts to look like the style of the style image. The final output is a new image containing the original content from the content image but is "painted" in the style of the image.

4 Numerical Experiments

4.1 Model parameters and learning curves

4.1.1 CycleGan

We followed the specifications given in the paper "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks"[1] to obtain the results and our implementation adopted the same set of parameters shown in table 1. For our analysis, we plotted the loss curves shown in Fig. 3; despite some fluctuations, the loss curves showed a general downward trend, which indicates the model was learning as intended. However, we limited the training to only 100 epochs due to the extensive time needed for training. We expect to see a more pronounced downward trend if we extend the number of epochs for the model.

Parameter	Value
Resolution	256*256
Batch Size	1
Epoch	100
Initial Learning Rate	0.0002
Adam beta1	0.5
Adam beta2	0.999
Adversarial Loss	MSE
Cycle Consistency Loss	L1 Loss
Identity Loss	L1 Loss
Cycle lambda	10
Identity lambda	0.5

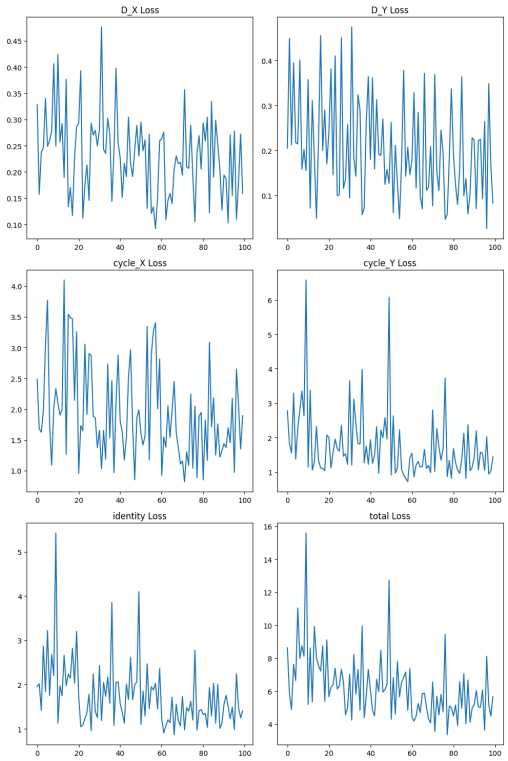


TABLE 1. CycleGAN Parameter Used in Training **FIGURE 3. CycleGAN Loss Curve**

4.1.2 NST

In Neural Style Transfer, there are many parameters that can influence the outcome; however, our team has found that the style image is the most impactful parameter among the parameters shown in Table 2. Therefore, the team will focus on demonstrating variations in the output solely by tuning the style images keeping all other parameters unchanged.

Parameter	Value
cnn	VGG19
cnn_normalization_mean	[0.485, 0.456, 0.406]
cnn_normalization_std	[0.229, 0.224, 0.225]

style_img	User-provided image
num_steps	500
style_weight	100000
content_weight	1

TABLE 2. NST Parameter

Now the team will illustrate the effects of varying style images on the outputs produced by NST. As observed in the results below in Table 3, the output image is significantly affected by the style image since the output image is trying to minimize the difference between the content image and the style image in terms of the style. By tuning the style image, our team has identified the third style image as the most effective in achieving the desired aesthetic. The team also plotted the loss curve for style as well as content losses in Figure 4, noting a consistent decrease that aligns with the diminishing disparity between the generated output and both style and content image, as anticipated.

Content Image	Style Image	Output Image
		
		
		

TABLE 3. NST Results by using different style image

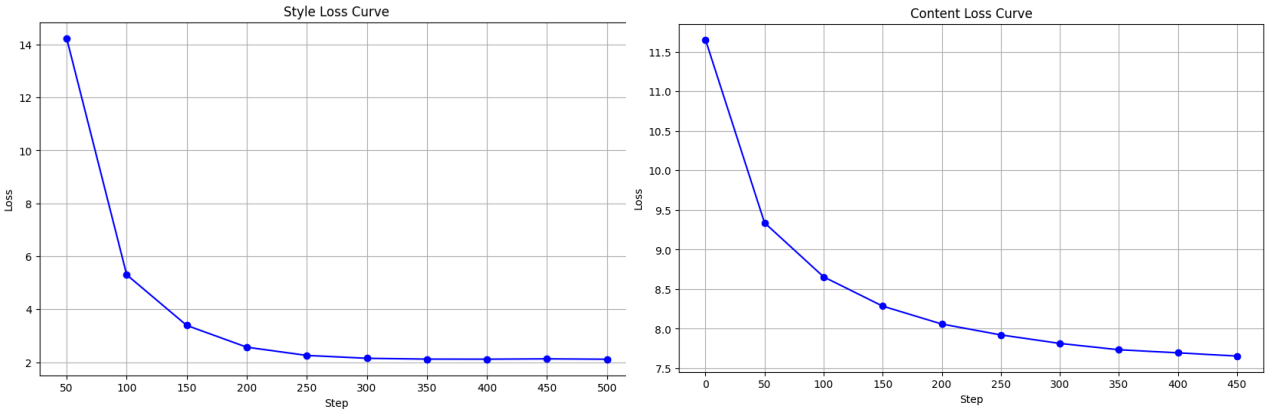


FIGURE 4: Style loss and Content loss for NST

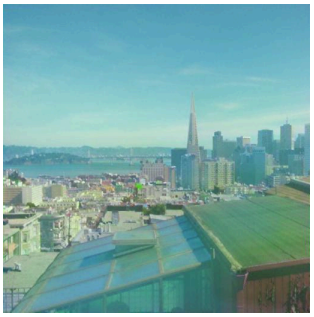
4.2 Numerical Evaluation Method:

4.2.1 Disadvantage of Fréchet Inception Distance method

Fréchet Inception Distance (FID)[3] metric is a quantitative measurement that is commonly used to assess the quality of the images that are generated by generative models. It quantifies the similarity between two sets of images, usually those of the generated and real images, by comparing the distribution of features computed by a pre-trained Inception v3 model.

However, FID will not capture the artistic nuances and fine detail of style that is unique to Ghibli artwork since it relies on features extracted by Inception v3, which is trained on ImageNet and may, therefore, not be very sensitive to stylistically different features spaces of Ghibli art. Generally, a lower FID score suggests greater similarity between the output and original images. This is a useful metric in quality assessment for images generated by models intending to produce images of high fidelity toward the input. However, when the purpose is only shifting toward the transformation of an image looking like it was drawn in Ghibli style, the fitness for use of the FID score becomes much less clear.

Examples: In Table 4, in the first example, img2, despite being transformed into the Ghibli style, got a higher FID score than the second example, which is the original image. If we rely solely on FID scores, we might incorrectly infer that the second scenario is better due to its lower score. However, this conclusion is incorrect since the second scenario lacks the Ghibli elements. This suggests the limitation of using FID scores as the quantitative metric for our style transformation tasks, as it fails to capture the essence of the desired stylistic conversion.

img1	img2	FID score
		144.184570231895

		0

TABLE 4. FID score for different images

4.2.2 Human Evaluation Method

Due to the limitation of the FID scores in assessing the transformation of images into Ghibli style, we adopted a human evaluation assessment instead. We developed a comprehensive rubric with four parts: Fidelity to the Ghibli Style, Preservation of Original Context, Emotional Resonance, and Coherence and Composition. Each aspect is scored on a binary scale, with a total possible score of 4 per image.

To conduct the evaluation, each team member rated a set of 20 images independently generated by the two models. We averaged the scores given by the evaluators for each image and then across each model to have an analysis for comparison between the models. In this way, we are able to assess not only the technical quality of style transfer but also the subjective, artistic attributes by which the Ghibli aesthetic is defined in a broader perspective with respect to our models for style transformation quality achievement. The results, detailed in Appendix B, reveal that the CycleGAN model achieved an average score of 2.975, while the NST (Neural Style Transfer) model obtained a lower average score of 2.1625.

Rubric for Rating the Style Transfer Tasks

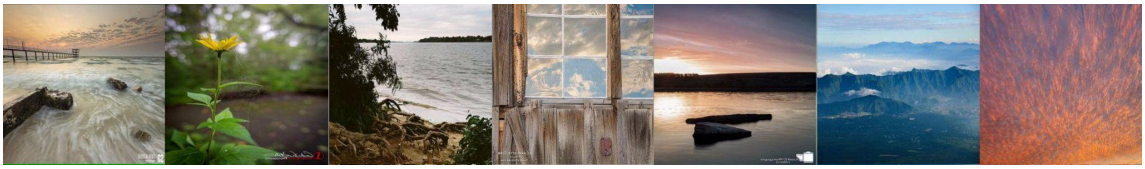
Aspects	Details	Score
Fidelity to Ghibli Style	Evaluate how well the output image presents the Ghibli art style in terms of color and texture.	0/1
Preservation of Original Context	Evaluate how well the revised image still represents the context and the essence of the original photograph by ensuring main elements and composition of the original are still recognizable and preserved.	0/1
Emotional Resonance	To what extent the image triggers feelings or moods typically provoked by Ghibli movies: feelings of wonder, nostalgia, or tranquility.	0/1
Coherence and Composition	Look at the overall composition and coherence of the picture. Everything blends flawlessly together in a way that is pleasant to both the aesthetic and the Ghibli narrative style.	0/1
Overall	Sum of the scores above	0-4

TABLE 5. Rubric for Human Evaluation

4.2.3 Comparison between two models

Sample output from two models

Original



CycleGan

NST

FIGURE 5: Outputs from the models

Based on the scores from the human evaluation results, the CycleGAN model clearly outperforms the NST model in the transformation to the Ghibli style. CycleGAN, considering its ability to learn from a great number of images, paid attention to even insignificant details with dignity, such as color and texture. For example, the clouds in the CycleGAN-generated images look quite like characteristic Ghibli-style clouds. However, the main disadvantage of CycleGAN is the training time, which took over 30 hours on this particular task. NST, on the other hand, gives relatively good results with very high speed, since it uses a pre-trained model. This approach eliminates the requirement for a very long training phase. However, the output quality from this style transfer network highly depends on the reference style image, so the performance of the model can vary a lot.

5 Conclusions

This project successfully fused the art style of Ghibli with real-world imagery using a CycleGAN model, advancing the realm of image style transfer in deep learning. The project highlighted the limitations of conventional quantitative assessments like FID in evaluating artistic transformations. Future work could focus on improving training efficiency, enriching the dataset, and crafting metrics that better capture the artistic style emulation. Overall, the project was implemented to broaden artistic style using artificial intelligence applications in creative industries.

References

[1]Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).

[2]Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.

[3]Fréchet inception distance. (2024, March 30). Wikipedia. https://en.wikipedia.org/wiki/Fr%C3%A9chet_inception_distance

[4]The Studio Ghibli Collection. (n.d.). The Studio Ghibli Collection. https://ghiblicollection.com/

Appendix A
CycleGAN paper: **https://arxiv.org/pdf/1703.10593.pdf**

Appendix B: Human Evaluation Results
CycleGAN human evaluation

CycleGan	Team member 1	Team member 2	Team member 3	Team member 4	
1	3	3	2	2	
2	2	2	3	4	
3	3	3	3	4	
4	3	4	3	3	
5	3	3	2	2	
6	3	3	4	4	
7	2	2	1	3	
8	4	4	3	4	
9	3	4	4	3	
10	3	3	3	3	
11	4	3	2	4	
12	3	2	2	1	
13	3	4	4	3	
14	3	2	3	4	
15	3	3	3	3	
16	3	4	2	2	
17	2	3	4	4	
18	2	2	2	3	
19	3	3	2	3	
20	4	4	3	4	
	2.95	3.05	2.75	3.15	2.975

Human evaluation for NST

NST	Team member 1	Team member 2	Team member 3	Team member 4	

1	3	3	2	2	
2	1	2	3	1	
3	2	3	3	3	
4	3	3	3	3	
5	0	2	1	2	
6	1	1	1	2	
7	2	2	3	3	
8	2	2	3	2	
9	1	2	0	1	
10	3	3	2	3	
11	3	3	2	1	
12	1	3	2	3	
13	3	2	3	2	
14	3	1	2	2	
15	2	2	3	2	
16	2	1	0	1	
17	3	2	3	4	
18	2	2	2	4	
19	3	2	0	3	
20	2	1	3	4	
AVG	2.1	2.1	2.05	2.4	2.1625