Image Synthesis using Generative Adversarial Networks

Kritika Ahuja, Ekta Goyal, Shikha Satsangi, C. Patvardhan

Dayalbagh Educational Institute, Agra, India

Corresponding author: Kritika Ahuja, Email: kritikaahuja2002056@dei.ac.in

In the past few years, significant progress has been made in artificial intelligence transforming many industries. Recently, the emergence and rapid adoption of advanced large language models like OpenAI's GPT, Google's PaLM, and Meta's Llama have shown great potential and sparked considerable global interest. However, there is a critical knowledge gap in the construction sector when it comes to the opportunities and challenges of implementing Generative AI, specifically in image-to-image generation tasks using Generative Adversarial Networks (GANs). By processing massive datasets of pictures, GAN models learn the essence of visual elements like color, shape, and color composition in the image. Within the world of artistic exploration, the generation of visual content opens the limitless potential for creativity. It also has the potential to generate an image in the style of Van Gogh or in the style of Raja Ravi Varma. Among these, CycleGAN has emerged as a powerful model for image-to-image translation tasks without requiring paired data. The following study explores the application of CycleGAN for preserving the content of input images while experimenting with various hyperparameters to enhance the model's performance. The objective is to assess the efficacy of CycleGAN in maintaining the integrity of the original content and compare its results against established tools such as NightCafe, Neural Style Transfer and Deep Dream Generator. The results of the following study indicate the that with careful tuning of hyperparameters, CycleGAN can achieve superior content preservation while generating visually appealing transformations. Furthermore, the comparative analysis reveals the strengths and limitations of each tool, highlighting CycleGAN's potential as a robust alternative for image transformation tasks.

Keywords: Style Transfer, GANs, CycleGAN, hyperparameter, Neural Style Transfer.

2024. In Mukesh Saraswat & Rajani Kumari (eds.), *Applied Intelligence and Computing*, 131–144. Computing & Intelligent Systems, SCRS, India. DOI: https://doi.org/10.56155/978-81-955020-9-7-15

1 Introduction

Generative AI refers to Artificial Intelligence where models are trained on a large dataset to create entirely new content. Generative AI can be used to generate text, images, videos, and even audio. Image generation not only allows us to generate anything from dreamlike art to photorealistic photos but also includes enhancing or inpainting the image. This technology extends to a vast range of applications, from enabling different artistic exploration to progress in scientific exploration, fundamentally evolving the way everyone experiences the visual world.
The proceeding of Generative AI to artificial intelligence is one more instance of the use of the human race's quest to instrumentalize machines with creative acumen, creativity, and invention (Prateek, 2024). Unlike trivial AI systems for classification, forecasting, or optimization methods done by people- the generative AI generates on its own. It creates novel content, paintings, themes, audio, and entire narratives and scripts. Generative AI models may include:

A. Text-to-text: It has automated numerous tasks that are concerned with text data, which has greatly improved natural language processing and human-computer interaction. Text-to-text generative AI uses artificial intelligence to generate text from text-based inputs. ChatGPT by OpenAI is a good example of text-to-text generative AI.
B. Text-to-Image: It allows for the generation/creation of vivid, lifelike images from a simple text description given as input, making even the most imaginative scenes visualizable.
C. Image-to-Image: Image-to-image translation is a technology that has been widely applied in different areas like computer vision, computer graphics, and image processing in recent years. The technology learns the features of the source pictures through neural network algorithms and transfers them to the output pictures. Some examples include a picture transferred into an oil painting style, a scene presented with RGB images, gradient fields, edge maps, semantic label maps, and so on (Li, 2022).
D. Text-to-Speech: This technology enables machines to convert written text into human-like speech. It also enables to creation of sound from the text-based input (NaturalReader).

In this paper, the focus will be on image-to-image generation. Image generation, a captivating branch of artificial intelligence, empowers machines to create entirely new images. This technology is rapidly transforming how interaction is done with visuals, from artistic creation to scientific exploration. Image-to-Image generation includes several techniques, one of the common techniques is Style Transfer.

Style Transfer is a prominent research area in computer vision that involves generating an image by combining the content of one image with the style of another image. It divides an image into a content image and a style image and then Style transfer is used to create an image that preserves the content of the original image when the visual style is applied to another image (Bethge, Gatys, Alexander, & Matthias, 2015). This process modifies the texture characteristics of the content image while preserving its semantic content, based on the texture synthesis/information of the style image (Cai, Ma, Wang, & Li, 2023).

How Style Transfer is better than Photoshop
Style transfer algorithms can apply art styles from one image to another, producing interesting results that mimic the art styles of a popular painting or graphic. This can be difficult and time-consuming to achieve in Photoshop. In Photoshop manual application of different styles is done on images while in style transfer this process can be automated by just training the model which is a less time-consuming process. Some style transfer algorithms enable image processing in real-time or near real-time which can be used for applications such as interactive artistic filters in mobile apps or live video processing. While in Photoshop complex edits take more computational resources and processing time. Style transfer offers experimentation with different styles while preserving the original content of the input image which might not be possible in the case of Photoshop. Style transfer using GAN/Generative AI can produce images with higher quality than the images created by using Photoshop. Generative AI enables professionals or people who are new in this field to create more diverse and interesting images

in less time. However, in Photoshop you might require some prerequisite knowledge/skill. Moreover, generative AI also offers to fine-tune the balance between the source image and the style image.

2 Methodology

As a first step towards the goal of Style transfer, the concept of Generative AI was explored. It is a form of Artificial Intelligence using which a variety of images, music, etc, can be produced. The following paper focuses on performing Style Transfer on the images. Different approaches were explored and understood for achieving Style Transfer. Neural Style Transfer is one such technique. It takes two images, one as the input image and the other as the style image. After blending both of these images, the output image contains the content of the input image and the style of the style reference image. Another approach for Style Transfer is Generative Adversarial Networks.

The initial encounter with GANs involve delving into its foundational structure and the intricacies of the adversarial training mechanism. Research has found that GANs have become increasingly popular for image generation and image-to-image translation. Its prominence stems from its proficiency in generative modeling, an area that revolves around the automated discovery and assimilation of patterns in input data, ultimately generating new examples that convincingly mirror those that could have originated from the initial dataset. On further exploration of GANs, it became evident that these models represent a cutting-edge approach within an area of deep learning for generative modeling. Its versatile applications extend to various domains, making it a powerful tool in tasks ranging from artificial image creation to sophisticated image-to-image translation processes and also for style transfer. Various types of GANs were inspected and it was figured that Cycle GAN was best suited for achieving the goal.

The detailed architecture of the Cycle GAN was understood and was then implemented. The results obtained from the Cycle GAN model are showcased in Section 7.2. Nowadays, there are several tools present for performing style transfer on the images, such as NightCafe Generator, Deep Dream Generator, etc. As a first step, these tools were experimented with for performing style transfer using a variety of styles (painting of different painters) along with the variation in the parameters. A brief overview of these tools is provided in Section 3 and the results obtained are presented in Section 7.1.

3 Existing Tools

There are various tools available for performing style transfer which can be listed as follows:

3.1 NightCafe

It is an AI art generator that uses multiple state-of-the-art machine-learning algorithms for image generation. It includes Stable Diffusion, Neural Style Transfer, DALL-E 2, CLIP-Guided Diffusion, and VQGAN+CLIP. This platform also provides different conversion styles to choose from and the ability of text-to-image translation. The platform facilitates the creation of striking art through two distinct approaches. Firstly, the traditional neural-style transfer method allows users to upload an image and select a "style" or reference image, guiding the AI to replicate the artwork (NightCafe).

3.2 Deep Dream Generator

Deep Dream Generator is an AI-powered platform that utilizes deep learning algorithms to transform images into surreal artworks. It allows to create/generate images from text-based prompts or image prompts (individually or in combination) (Hai-Jew, 2023).

4 Methods/Architecture

4.1 Neural Style Transfer

Neural Style Transfer is one of the methods by which style transfer in images can be achieved. In this technique, two input images are taken – one is the image on which the style transfer is being applied (Image A) and another one is the image of the whole style which is to be imposed (Image B). These two images blend with each other and produce the output image which contains the content of Image A painted in the style of Image B. The output image is produced by matching the content statistics of Image A and the style statistics of Image B. Convolutional Network is used for extracting these statistics from Image A and Image B (Systems).

How does Neural Style Transfer Image work
Convolution Neural Networks are used for performing style transfer in the images. In CNN, different layers are present. Each layer has number of filters depending upon the number of layers. These filters help in the extraction of the content and style statistics of the image. CNNs are not aware of the content of the image, they encode from the image and learn from it, which helps in Neural Style Transfer.
The process involved in style transfer begins with the initialization of a noisy image (C) which will be the output image. This image is then compared to the content image (A) and the style image (B) and the similarity with these two is being calculated at the particular layer of the VGG network. The loss of C with respect to A (Content loss) and B (Style loss) is then calculated. This total loss produced is tried to minimize as much as possible through backpropagation, leading to more optimized output images.
The losses involved in the Neural Style Transfer are as follows:

A. **Content Loss -** Content loss implies how much the generated image C is similar to that of the image A (Shaw, 2023).

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2}\sum_{i,j}\left(F_{ij}^{l} - P_{ij}^{l}\right)^2 \quad (1)$$

$\vec{p}$	The original image
$\vec{x}$	The generated image
l	Layer
F_{ij}^{l}	Activation of the ith filter at position j in the feature representation of $\vec{x}$ inl
P_{ij}^{l}	Activation of the ith filter at position j in the feature representation of $\vec{p}$inl

B. **Style Loss -** Style loss can be understood as the squared difference between the Gram Matrix of Image A and the Gram Matrix of Image C.

$$\mathcal{L}_{style} = \sum_{l}\sum_{i,j}\left(G_{i,j}^{s,l} - G_{i,j}^{p,l}\right)^2 \quad (2)$$

$G_{i,j}^{s,l}$	Gram Matrix of the style image
$G_{i,j}^{p,l}$	Gram Matrix of the generated image

C. **Total Loss Function -** The total loss function is the sum of the content and the style loss.

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha\mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta\mathcal{L}_{style}(\vec{a}, \vec{x}) \quad (3)$$

α, and β are used for weighting Content and Style cost respectively. They define the weightage of each cost in the generated output image (Singh, 2019).

4.2 GANs

Generative Adversarial Networks or GANs are deep learning models (neural networks, similar to that of the human brain) that try to learn complex patterns from the data (images, sound, text, etc). A GAN model usually utilizes two sets of neural networks, the Generator and the Discriminator. A generator tries to generate the images close to that of the input image, whereas the discriminator compares the output generated by the generator and the input image, followed by providing feedback to the generator. In short, it can be understood as the generator tries to fool the discriminator by generating images similar to that of the input image and the discriminator tries to identify the fake image generated by the generator. By this adversarial training, both of the networks try to improve their ability. Both the generator and the discriminator are trained separately (Barla, 2024).

The loss function as computed after the role of the discriminator is as follows:

min G max D_G V (D, G) = Ex [log D_G (x)] +Ez [log (1 – D_G, (G(z))] (4)

In the above expression, the generator tries to minimize this part and the discriminator tries to maximize this part. In this manner, a min-max game takes place between the discriminator and generator, and both the models get trained (Zhu, Park, Isola, & Efros, 2020).

Cycle Gan Architecture
Like other GANs, Cycle GAN also consists of Generator and the discriminator. However, here there are two generators and discriminators (Johnson, Alahi, & Fei-Fei, 2016). The architecture of Generator and Discriminator of the Cycle GAN has been showcased below.(Figure 1 and Figure 2)

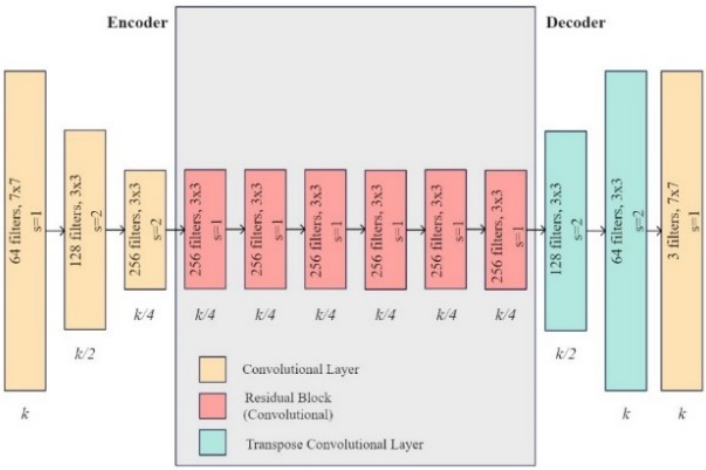


Figure 1. Generator architecture of CycleGAN

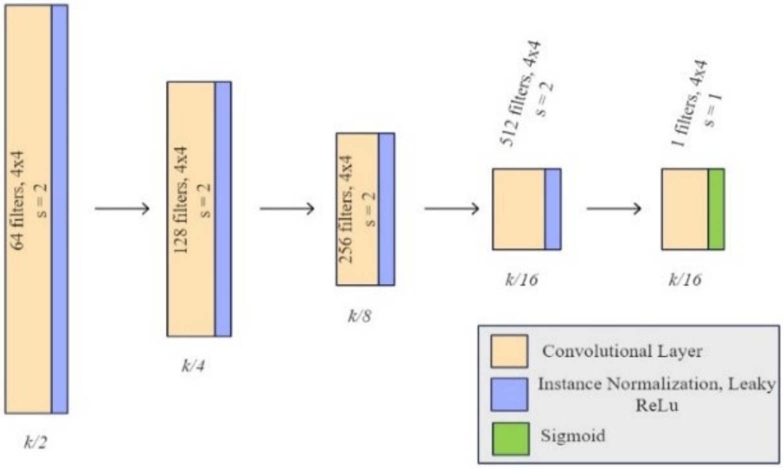


Figure 2. Discriminator Architecture of CycleGAN

- **Architecture of Generator in Cycle GAN:** Generator in Cycle GAN majorly consists of encoder, transformer, and decoder.
- **Architecture of Encoder in Generator:** The input given to the generator first passes through the encoder which extracts the features from the input image through Convolutions (3), followed by the compression of the image (up to 1/4th of the actual image) along with the increase in the number of channels. The activation function is then applied to the output of the encoder (feature volume) and passed to the transformer.
- **Architecture of Transformer:** The transformer consists of the Resnet blocks. The feature volume is then passed through these Resnet blocks (6). Each Resnet block consists of 2 convolution layers with a by-pass. This by-pass in Resnet block allows the transformation of the earlier layers to be retained from the previous layers, thus allowing to build deeper networks.
- **Architecture of Decoder in Cycle GAN:** Decoder utilizes the feature volume provided by the transformer and converts into the output image. It consists of 2 deconvolution layers which give the image and then this image is passed through the final convolution layer to give the output image.
- **Architecture of Discriminator in Cycle GAN:** The discriminator utilizes the Patch GAN as its architecture. It involves cropping the image then the results are averaged. Using this, it is determined the image belongs to which category, i.e., is it real or fake.
- **Working of the cycle GAN and Losses involved in the Cycle GAN:** To understand the working of the cycle GAN, let us consider here two domains, X and Y, where our goal is to perform the style transfer from domain X to domain Y. X consists of the images from its domain while Y contains images of its own. The images in X and Y are unpaired here.(Figure 3)

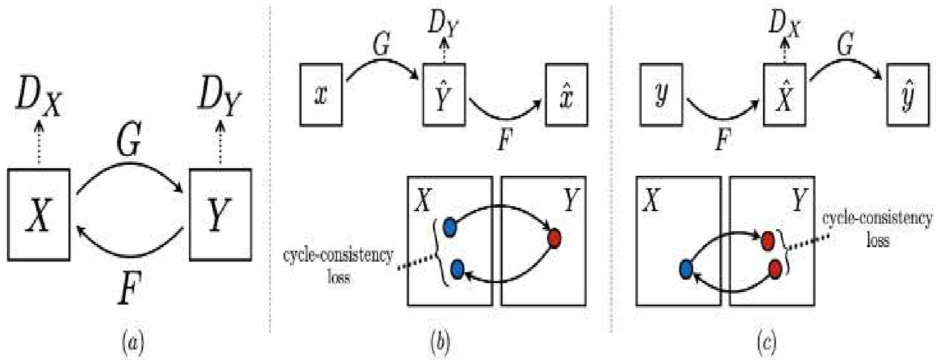


Figure 3. (a) The Cycle GAN model contains two mapping functions G: X -> Y and F : Y-> X and associated adversarial discriminators, $DYandDX$, $.DY$ encourages G to translate X into outputs indistinguishable from domain Y and vice versa for DX, and F. To further regulate the mappings, two cycle consistencies loss have been introduced that captures the intuition that if translation is done from one domain to another and back again, the output should resemble with that of the input. (b) Forward Cycle Consistency Loss: x-> F(G(x)) ~ x , and (c) backward cycle consistency loss y-> F(y) -> G(F(y)) ~y (Zhu, Park, Isola, & Efros, 2020)

Two types of losses play a key role in the style transfer from X to Y.

Adversarial Loss

The input image in given to the generator G which produces an image of the form (G(x)), i.e. G: X->Y. Now, the discriminator distinguishes whether it (G(x)) is fake or real (belongs to the Y domain or not) with the help of Adversarial Loss. This loss helps in the training of the working of the generator and discriminator. (Nanos, 2024)

Moreover, there is another generator F, which produces an image of the form (F(y)), i.e. F: Y->X. This image is passed to the discriminator, which checks if the output image (F(y)) belongs to domain X or not, through Adversarial loss. Thus, it can be seen that adversarial loss is being utilized in both generator-discriminator models. For the G:X->Y, the Adversarial loss utilized, is as follows:

$$\mathcal{L}_{GAN}(G, D_G, , X, Y) = E_{y\sim pdata(y)}[\log D_G, (y)] + E_{x\sim pdata(x)}[\log(1 - D_G, (G(x))] \quad (5)$$

In the above loss function, the G tries to minimize the loss in the generation of the G(x), whereas, tries to maximize the loss. Thus, this min-max game takes place here, or in other words, it can be summed as,

$$min_{G} \max D_G \mathcal{L}_{GAN}(G, D_G, , X, Y) = E_{y\sim pdata(y)}[\log D_G, (y)] + E_{x\sim pdata(x)}[\log(1 - D_G, (G(x))] \quad (6)$$

Similarly, the adversarial loss for the second generator and discriminator model is as follows,

$$min_{F} \max D_F \mathcal{L}_{GAN}(F, D_F, , X, Y) = E_{x\sim pdata(x)}[\log D_F, (x)] + E_{y\sim pdata(y)}[\log(1 - D_F, (G(y))] \quad (7)$$

Cycle Consistency Loss
However, the adversarial loss is not sufficient in the Cycle GAN as the data utilized here in the domain X and Y are unpaired here and the same set of images in domain X can map to any random permutations of images in the domain Y, leading to the similarity in the output and the target distribution. To overcome this problem in Cycle GAN, it was proposed that the functions obtaining a mapping from domain X to Y should be cycle-consistent, i.e.

$$x \rightarrow G(x) \rightarrow F(G(x)) \approx x \quad (8)$$

The above cycle is known as the forward cycle consistency, whereas the reverse cycle consistency is as follows.

$$y \rightarrow F(y) \rightarrow G(F(y)) \approx y \quad (9)$$

The cycle consistency is achieved with the help of the Cycle Consistency Loss, which is as:

$$\mathcal{L}_{Cyc}(G, F) = E_{x\sim pdata(x)}[\|F(G(x)) - x\|_1] + E_{y\sim pdata(y)}[\|G(F(y)) - y\|_1]. \quad (10)$$

The overall objective however in Cycle GAN is as:

$$\mathcal{L}(G, F, D_G, D_F) = \mathcal{L}_{GAN}(G, D_G, \ , X, Y) + \mathcal{L}_{GAN}(F, D_F, X, Y) + \lambda\mathcal{L}_{Cyc}(G, F) \quad (11)$$

Here is used for controlling the importance of the two aims, i.e.

$$G^*, F^* = \arg \min_{G,F} \max_{D_G, D_F,} \mathcal{L}(G, F, D_G, D_F) \quad (12)$$

5 Dataset Information and Processing

For transferring the style of Raja Ravi Varma, 227 images of paintings of Raja Ravi Varma and 6287 natural images were utilized. Majorly, the dataset of the Raja Ravi Varma has been formed through the official site of his collection, The Raja Ravi Varma Heritage Foundation. The dataset of the natural images was obtained from the Kaggle.

The dataset is trained for 100 epochs at a learning rate of 0.0002. The generator filter in the last convolutional layer and the discriminator filter in the first convolutional layer is 64. The discriminator architecture used is a 70x70 patchGAN and 9 residual blocks are used in generator architecture.

6 Performance Metrics

LPIPS (Learned Perceptual Image Patch Similarity): LPIPS essentially computes the similarity between the activations of two image patches for some pre-defined network. This measure has been shown to match human perception well. A low LPIPS score means that image patches are perceptual similar.

This metric leverages deep learning to compare images based on how similar their features appear to a pre-trained convolutional neural network (CNN). It aims to capture how humans perceive image similarity, making it strong for tasks like image generation and editing where perceptual quality matters most. Lower LPIPS scores indicate greater similarity.

PSNR (Peak Signal-to-Noise Ratio): This metric is a traditional method that calculates the difference between corresponding pixels in two images. Higher PSNR indicates a lower difference and supposedly better quality. However, PSNR can be fooled by artifacts that don't affect human perception much.

SSIM (Structural Similarity Index Measure): This metric goes beyond just pixel differences. It considers factors like luminance, contrast, and structure to measure image similarity. SSIM values closer to 1 indicate greater similarity.

7 Result

In this study, the compelling outcomes were observed that highlight the efficacy of the existing tools, Neural Style Transfer and CycleGAN. Leveraging well-established tools in the field, impressive image synthesis with a high degree of realism and diversity were achieved. The images generated using CycleGAN exhibit intricate details, nuanced textures, and vivid colors, showcasing the GANs' capability to capture complex patterns in the training data. The experiments also revealed the influence of hyperparameter tuning and model architecture on the final output, providing valuable insights for future implementations. Overall, the results underscore the potential of employing existing GAN tools for image generation tasks, emphasizing their adaptability and effectiveness in producing visually appealing and contextually relevant content.

7.1 Results of Style Transfer using Neural Style Transfer and Existing tools

The content image used for style transfer is given in Figure 4.

Figure 4. Content Image

Table 1. Results of Style Transfer using Neural Style Transfer and Existing Tools along with performance metrics

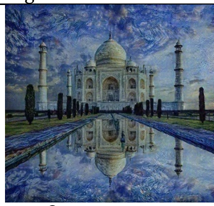
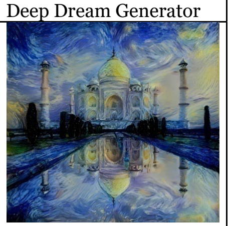
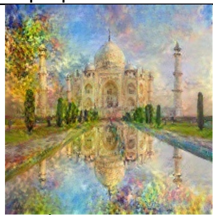
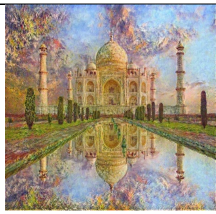
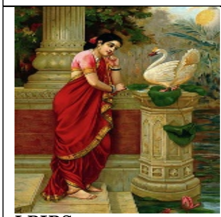
Style	Neural Style Transfer	NightCafe	Deep Dream Generator
 LPIPS PSNR SSIM	 [illegible] 27.6846 0.4214	 0.5380 27.6089 0.32308	 0.5274 27.6304 0.3800
 LPIPS PSNR SSIM	 0.74651 27.9567 0.4730	 0.5059 27.7836 0.3239	 0.3598 27.9914 0.6294
 LPIPS: PSNR: SSIM:	 0.6832 27.7339 0.4994	 0.5797 27.8329 0.4491	 0.7239 27.5882 0.5224
 LPIPS PSNR SSIM	 0.7473 27.7250 0.4410	 0.6389 27.7187 0.3096	 0.5040 28.1055 0.5006

Table 1. represents the results obtained using Neural Style Transfer and existing tools (NightCafe and Deep Dream Generator). The input/content image for all the experiments in each tool is shown in Fig 4. The first column of table represents the style of the image that has to be transferred to the input/content image. The second, third and fourth column of the table displays the result obtained using Neural Style Transfer, NightCafe and Deep Dream Generator respectively.

Discussion

The distinct artistic styles of the following artists were used for the experimentation:

- Van Gogh's "Starry Night",
- Claude Monet's "Villas at Bordighera", and
- Raja Ravi Varma's "Hamsa Damayanti" and "Vasantika".

When applying the style of "Starry Night" to the Taj Mahal image, using NightCafe, the results were striking. The output image yielded vibrant, swirling patterns reminiscent of the input image. The colors appeared richer, with pronounced brushstrokes; while Deep Dream Generator exhibited an image with hallucinatory patterns and textures overlaying the structure. However, the details of the monument seemed somewhat obscured by the intense stylization along with some distortion observed was curved or swirl patterns along the perimeter of the Taj Mahal.

When applying the style of "Hamsa Damayanti" to the Taj Mahal image, using NightCafe, the resulting image showcased a blend of realism and surrealism, with elements of Raja Ravi Varma's distinctive aesthetic superimposed onto the Taj Mahal. The intricate detailing and warm color palette characteristic of Raja Ravi Varma's paintings transformed the Taj Mahal into a scene evocative of classical Indian artwork. The Deep Dream Generator resulted in a similar output image as that of NightCafe. However, upon closer examination, subtle differences emerged between the outputs generated by Deep Dream Generator and NightCafe. In the case of Deep Dream Generator, the detailing appeared a bit obscured, with finer nuances and intricacies of Raja Ravi Varma's artwork less pronounced. Additionally, there was a noticeable smoothing effect applied to the output image, resulting in a more uniform texture and surface appearance.

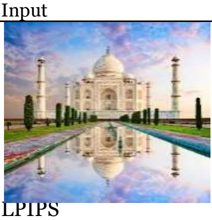
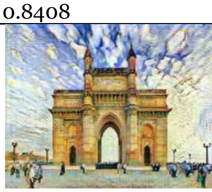
When applying the style of "Villas at Bordighera" to the Taj Mahal image using NightCafe, the results were striking and mesmerizing. The vibrant brushstrokes and vivid color palette characteristics of Monet's work transformed the Taj Mahal into a scene reminiscent of the French Riviera. The soft, diffused lighting and dappled shadows evoked a sense of atmospheric depth, imbuing the monument with a serene, almost ethereal quality. The architectural details of the Taj Mahal remained discernible while using Deep Dream Generator, the details were enveloped in a veil of impressionistic abstraction, blurring the boundaries between form and color. Though the colors became more vibrant using the Deep Dream Generator but the brushstrokes and the texture did not resemble with Claude Monet's artwork.

While applying the style of "Vasantika" to the Taj Mahal image. The NightCafe attempted to infuse the input image with the essence of Raja Ravi Varma's classical Indian aesthetic, the outcome fell short of expectations. Instead of a harmonious fusion of artistry and architecture, the resulting image exhibited notable distortions and blurring of details. The Taj Mahal, enveloped in Ravi Varma's style, suffered from a loss of structural integrity, with architectural elements appearing warped and disproportionate. Additionally, the intricate detailing characteristic of Ravi Varma's artworks was obscured, replaced by a muddled depiction that lacked clarity and definition. Moreover, artifacts from the style reference image, such as flowers, were inadvertently transferred onto the Taj Mahal image, further detracting from its fidelity and coherence. The Nightcafe successfully replicated the vibrant colors of the style reference image, while the Deep Dream Generator the resulting image exhibited minimal stylization, characterized primarily by a smoothing effect applied to the overall composition.

The output images obtained using neural style transfer are similar to images obtained using existing tools mentioned above. The resultant images having the style of "Starry Night" by Van Gogh and "Villas at Bordighera" by Claude Monet are similar to images obtained using Deep Dream Generator tool while the resultant images having the style of "Hamsa Damayanti" and "Vasantika" by Raja Ravi Varma resembles to the images obtained using NightCafe tool.

7.2 Results of Style Transfer using CycleGAN

Table 2. Results of Style Transfer using CycleGAN with performance metrics

Input	Monet	Van Gogh	Raja Ravi Varma
 LPIPS PSNR SSIM	 0.6809 28.2081 0.5825	 0.5510 28.3779 0.6673	 0.4563 28.1903 0.6260
 LPIPS PSNR SSIM	 0.1667 32.2653 0.9065	 0.2000 32.3242 0.9110	 0.1429 29.0019 0.7313
 LPIPS PSNR SSIM	 0.2708 27.6346 0.8100	 0.4566 28.1191 0.6994	 0.4941 28.2404 0.7041
 LPIPS PSNR SSIM	 0.1699 28.7912 0.9129	 0.3109 28.4147 0.8408	 0.5721 28.0931 0.7041
 LPIPS PSNR SSIM	 0.1992 28.4971 0.9148	 0.4075 28.7724 0.8208	 0.4420 28.3116 0.7252

The table 2 showcases the results of the Cycle GAN model. The style transfer is performed for the various artists including Van Gogh, Monet and Raja Ravi Varma. The first column of the table represents the input image given to the Cycle GAN model. The second column showcases the output images in the style of Van Gogh, third column represents them in the style of Monet while fourth column showcases the images in the Raja Ravi Varma's style. The performance metrics values for all images are better than the performance metrics values of existing tools.

Discussion:
Unlike Existing Tools (NightCafe and Deep Dream Generator) and Neural Style Transfer, Cycle GAN trains on multiple images and learns the pattern along with the style. Moreover, it transfers the style to the input image, whereas Existing tools and Neural Style Transfer transfers one image's style to another. Raja Ravi Varma, the Indian artist and painter blended European realism with Indian sensibilities in his artworks/paintings. Raja Ravi Varma's realistic style describes figures, and scenes with accuracy and details. His paintings used a rich color palette to enhance the visual impact. His paintings use softer, pastel shades incorporating vibrant colors to draw attention.

8 Advantages and Limitations

- Advantages of CycleGAN
  - It works well for image-to-image generation tasks involving transfer of style/elements.
  - It is bidirectional, it can convert to and from the input and style image.
  - It trains without the need of paired data, making it applicable to a wide range of tasks where paired data is scarce or unavailable.
  - It is flexible and adaptable.
  - It focuses on learning the mapping between two domains, rather than between two specific images.

- Limitation of CycleGAN
  - Difficult to handle large domain gaps, it, may result in less accurate translation.
  - Sensitive to hypermeters and network architecture.
  - CycleGAN does not perform very well when applied to perform geometrical transformation, because of the generator architecture which is trained to perform appearance changes in the image.
  - It doesn’t take image supported prompts for generating new image.

9 Conclusion

In conclusion, the exploration of image generation using Generative Adversarial Networks (GANs) through existing tools provides valuable insights into both the capabilities and limitations of this cutting-edge technology. The results obtained showcase the potential of GANs in creating visually striking and diverse images, demonstrating their capacity to learn intricate patterns from training data. Despite the impressive outcomes, challenges such as mode collapse and the presence of artifacts highlight the need for ongoing advancements in GAN architectures and training methodologies. Additionally, the reliance on existing tools prompts a critical consideration of the trade-offs between ease of use and the potential for customization. Future research endeavors should aim to strike a balance, leveraging the strengths of established tools while addressing their inherent constraints. As the field of GANs continues to evolve, these findings underscore the importance of refining existing frameworks and developing novel approaches to unlock the full potential of GANs in image generation applications.

References

[1] Prateek, "Generative AI Tutorial for Beginners: A Comprehensive Guide," 27 April 2024. [Online]. Available: https://www.edureka.co/blog/generative-ai-tutorial/#Image.

[2] X. Li, "Study on Image-to-image Translation Based on Generative Adversarial Networks," in 2022 2nd International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR), 2022, pp. 92-97.

[3] "NaturalReader," [Online]. Available: https://www.naturalreaders.com/online/.

[4] Bethge, L. A. Gatys, S. E. Alexander and Matthias, "A Neural Algorithm of Artistic Style," 2015.

[5] Q. Cai, M. Ma, C. Wang and H. Li, "Image neural style transfer: A review," Computers and Electrical Engineering, vol. 108, p. 108723, 2023.

[6] NightCafe, "Create Stunning AI Art," Night Cafe, [Online]. Available: https://creator.nightcafe.studio/.

[7] S. Hai-Jew, "Exploring the Deep Dream Generator (an Art-Making Generative AI)," Sildeshare, 4 April 2023. [Online]. Available: https://www.slideshare.net/ShalinHaiJew/exploring-the-deep-dream-generator-an-artmaking-generative-ai.

[8] T. L.-S. M. L. o. H. Systems, "An end-to-end platform for machine learning," TensorFlow, [Online]. Available: https://www.tensorflow.org/.

[9] S. Shaw, "A Complete Step-wise Guide on Neural Style Transfer," Medium, 25 March 2023. [Online]. Available: https://medium.com/@shwetaka1988/a-complete-step-wise-guide-on-neural-style-transfer-9f60b22b4f75.

[10] A. Singh, "How Do Neural Style Transfers Work?," Medium, 19 July 2019. [Online]. Available: https://towardsdatascience.com/how-do-neural-style-transfers-work-b76de101eb3.

[11] N. Barla, "Generative Adversarial Networks and Some of GAN Applications: Everything You Need to Know," neptune.ai, 24 April 2024. [Online]. Available: https://neptune.ai/blog/generative-adversarial-networks-gan-applications.

[12] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," in IEEE international conference, 2020.

[13] J. Johnson, A. Alahi and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," 2016.

[14] G. Nanos, "Generative Adversarial Networks: Discriminator's Loss and Generator's Loss," baeldung, 18 March 2024. [Online]. Available: https://www.baeldung.com/cs/gan-loss.