Neural Style Transfer

Nandini Jaiswal

MDS202335

Nandini Jaiswal

Introduction

Neural Style Transfer (NST) is a fascinating deep learning application where the style of one image is combined with the content of another, creating a new image that preserves the subject of one while mimicking the artistic style of the other. Introduced by Gatys et al. in their seminal 2015 paper, NST leverages the power of convolutional neural networks (CNNs) trained on object recognition tasks to extract and recombine these two independent elements: content and style.

Objective

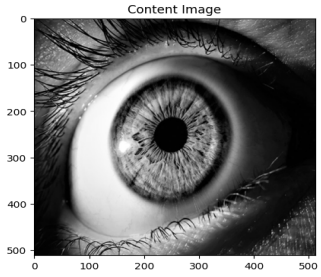
The goal of this project is to implement a Neural Style Transfer (NST) model that blends the artistic styles of multiple images with the content of another image. The approach leverages a pretrained VGG19 model to optimize an output image that minimizes style and content loss, creating a visually striking fusion of style and content.

Dataset Overview

Neural Style Transfer uses images for two distinct roles:

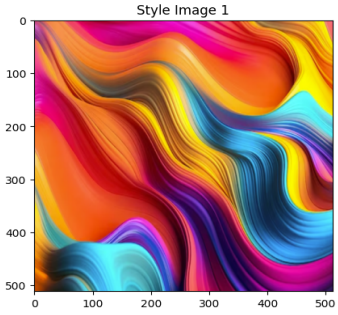
1. *Content Images:* The primary image whose structural layout and main elements are preserved.

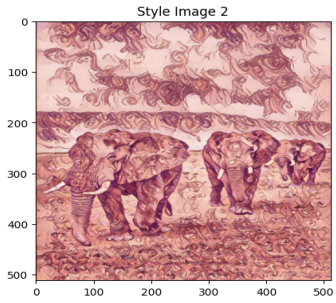
    Example: A photograph of an eye, Face of a human.



2. *Style Images:* One or more images that serve as the source of artistic texture and patterns.

    Example: Artwork or design images with unique stylistic elements.





These images are resized and converted into tensors compatible with PyTorch.

Technical Foundation

1. Content and Style Representation

NST utilizes pre-trained CNNs like VGG-19, specifically convolutional layers, to extract hierarchical feature representations:

- **Content Representation:** Extracted from deeper layers of the CNN, where high-level features of the image (e.g., shapes and objects) are encoded.
- **Style Representation:** Modeled using correlations between feature maps, encapsulated in Gram matrices. These matrices capture texture and patterns like brushstrokes or color distribution, making them ideal for style characterization.

2. The Loss Function

NST relies on an optimization problem driven by a composite loss function:

- **Content Loss:** Measures the difference between the content feature representations of the input image and the target content image.
- **Style Loss:** Quantifies the difference in Gram matrices between the input and the target style images.
- **Total Loss:** A weighted sum of the content and style losses, controlled by hyperparameters α and β.

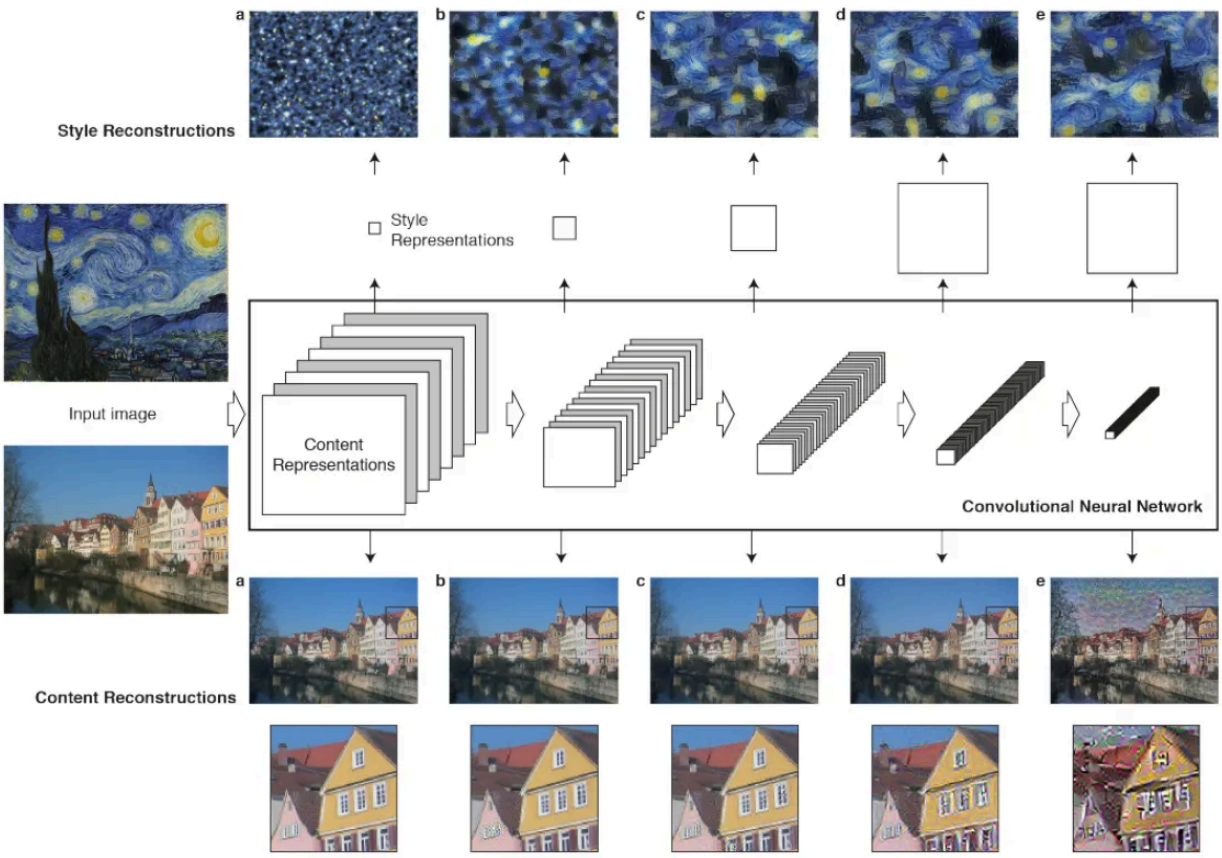
3. Optimization

The process involves starting with a random noise image and iteratively updating it using backpropagation to minimize the total loss, thereby creating the stylized output.

Visualizing Style and Content

In the process of Neural Style Transfer, convolutional neural networks (CNNs) play a vital role in extracting and representing the key features of both content and style images. Content representations capture the structural elements of the input image, while style representations focus on the artistic patterns, textures, and colors.

The figure below illustrates the reconstruction process for both style and content. It demonstrates how different layers of a CNN contribute to isolating style features from a style image (such as Van Gogh's Starry Night) and content features from a content image. It also highlights the reconstructed outputs at various levels of the network, showcasing the separation and recombination of style and content to achieve the artistic transformation.



Data Inspection

Image Preprocessing:

- Images are loaded and resized to a fixed dimension (128x128 for CPU and 512x512 for GPU environments).

- A normalization step ensures compatibility with the VGG19 network by standardizing RGB pixel values to predefined mean and standard deviation.

Image Visualization:

- Both content and style images are displayed for inspection using matplotlib.
- This ensures the preprocessing steps are verified visually.

Model Design

The model builds upon the pretrained VGG19 architecture to calculate and minimize two key metrics:

1. Content Loss: Measures the similarity between the output image and the content image. Uses Mean Squared Error (MSE) to compare feature maps of both images at selected convolutional layers.

2. Style Loss: Measures the similarity between the output image and the style images. Uses the Gram matrix to represent the correlation of feature maps and computes the MSE between Gram matrices of the style and output images. Incorporates weights to balance the contribution of multiple style images.

3. Normalization Layer: A custom normalization layer is added to ensure input images align with the VGG19 network's expected scale.

Feature Extraction Layers:

Style Layers: Extract features from deeper layers (**conv_1**, **conv_2**, **conv_3**, **conv_4**, **conv_5**).

Content Layer: Focuses on features from the **conv_4** layer.

Model Initialization And Training

1) Initialization:

- A clone of the content image is used as the starting point for optimization.
- Content and style features are extracted from respective layers of the pretrained **VGG19** model.

2) Optimizer:

- The **LBFGS optimizer** is used to iteratively update the output image.
- The output is constrained within valid pixel ranges using clamping techniques.

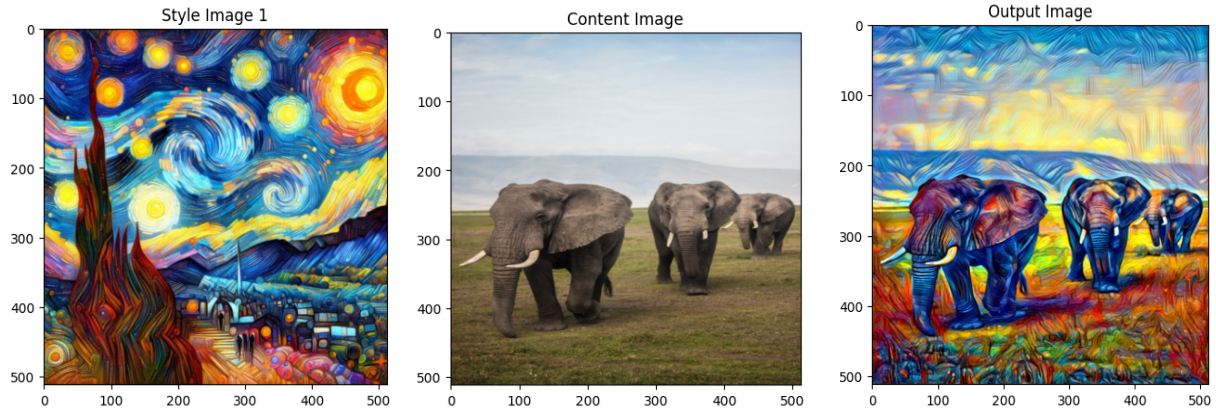
3) Training Objective:

- Minimize the combined loss function:
- Total Loss = $\alpha \cdot$ Content Loss + $\beta \cdot$ Style Loss

- Content weight (α) = 1
- Style weight (β) = 100,000
- Optimization is run for 300 steps with progress logged at intervals.

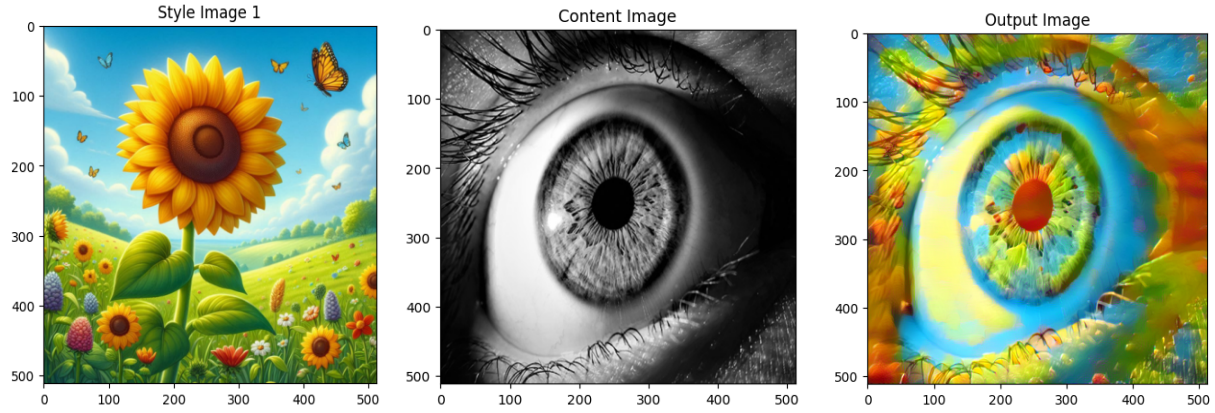
Results And Discussion

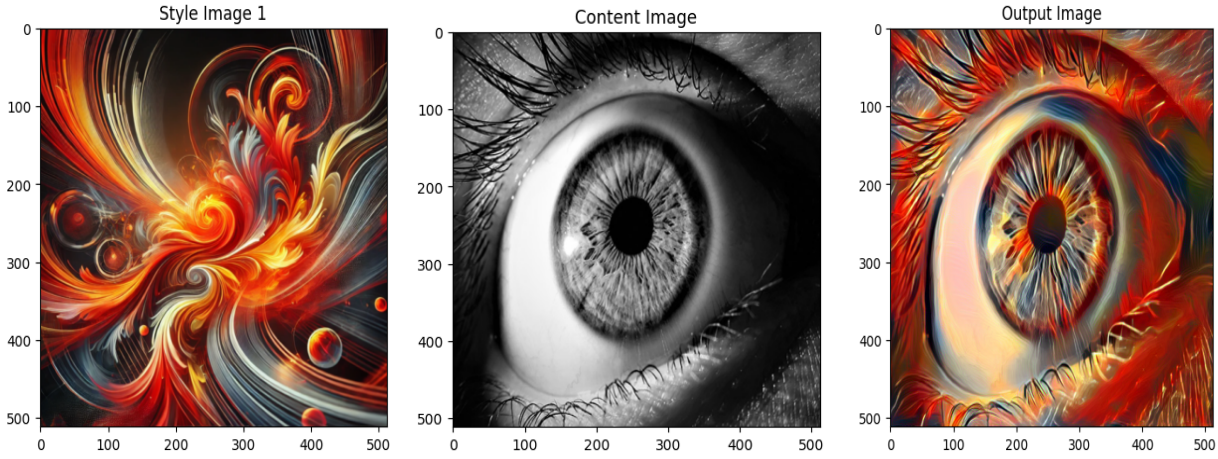
Below are some of the visual results achieved through the algorithm, reflecting the influence of the style images on the content image.

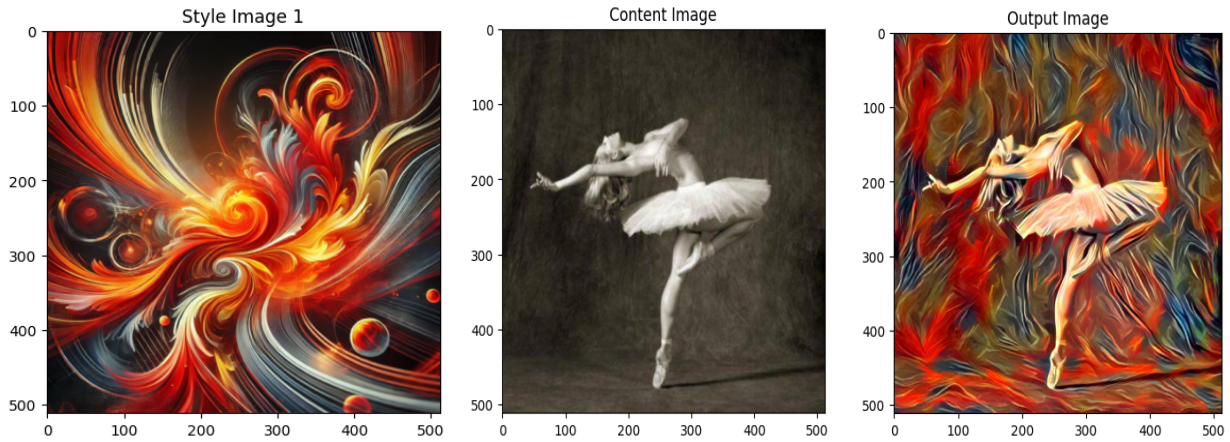


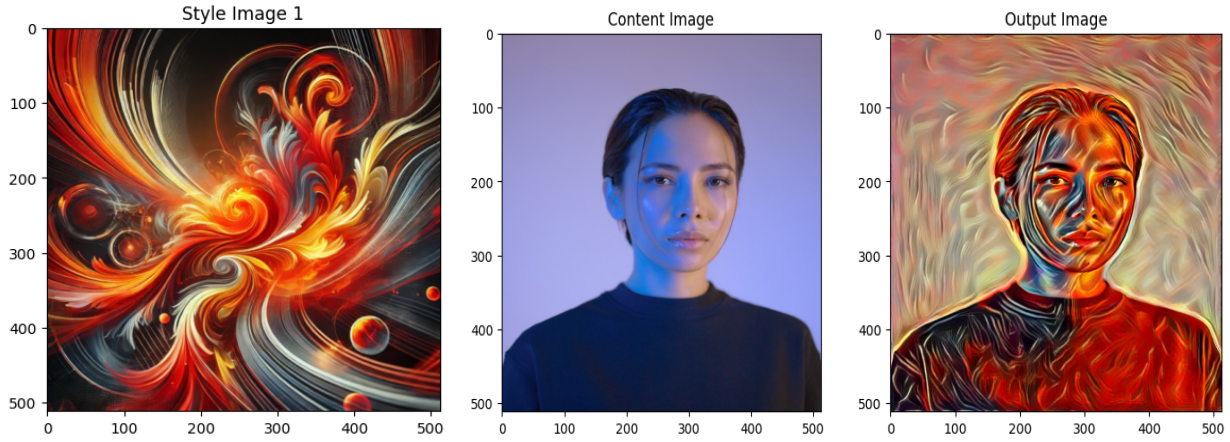
Above sequence of images demonstrates the style transfer from Vincent Van Gogh's beautiful 'The Starry Night' painting to two different content images.

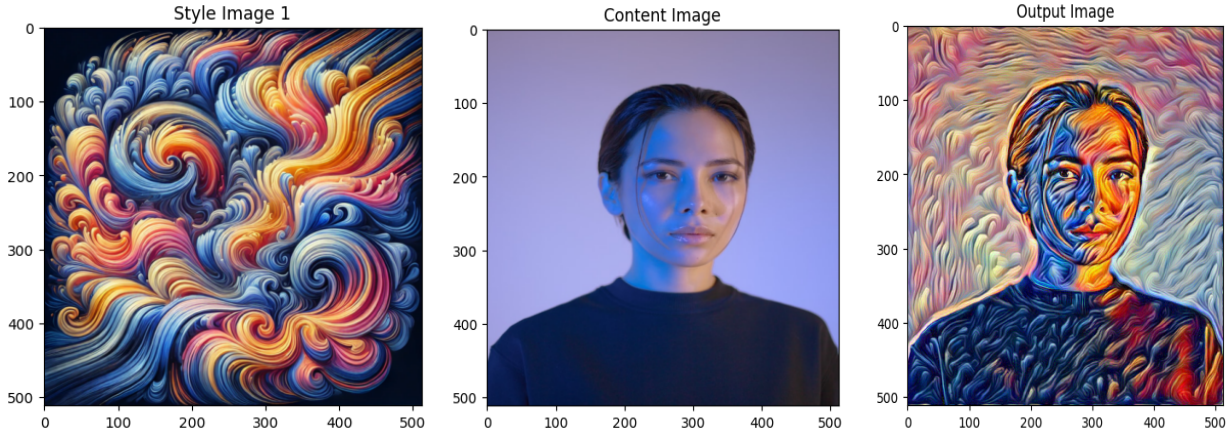
The final output images are a combination of the structural features of the content image with the artistic patterns of the style images.

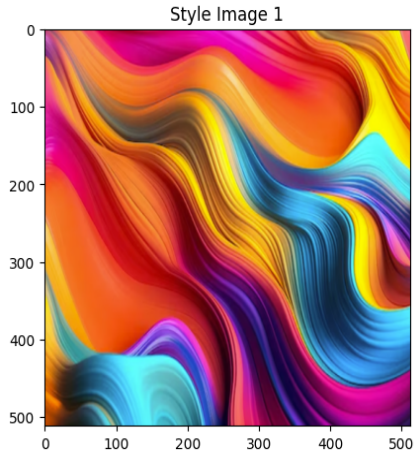


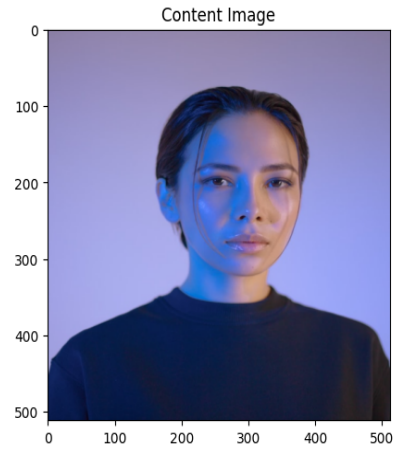


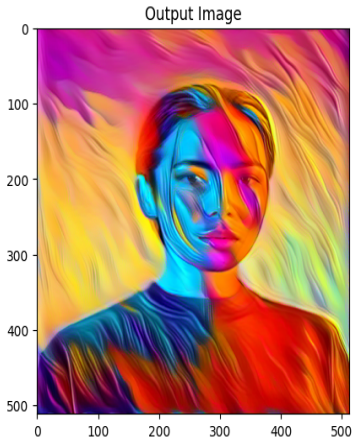

Style Image 1
Content Image
Output Image

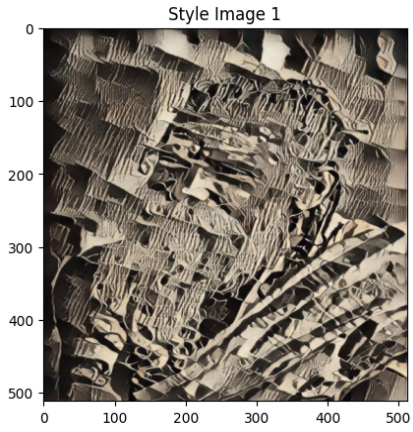

Style Image 1
Content Image
Output Image


Style Image 1
Content Image
Output Image

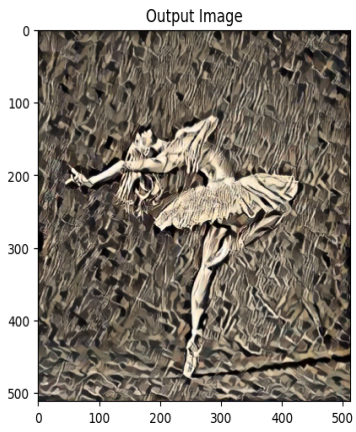

Style Image 1

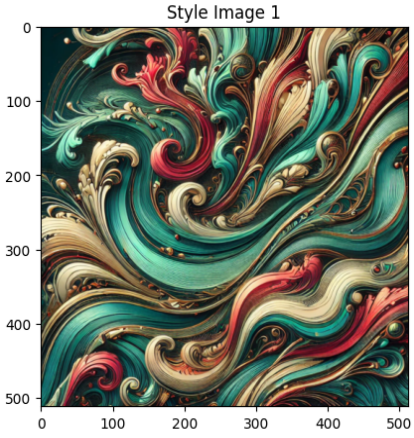

Content Image

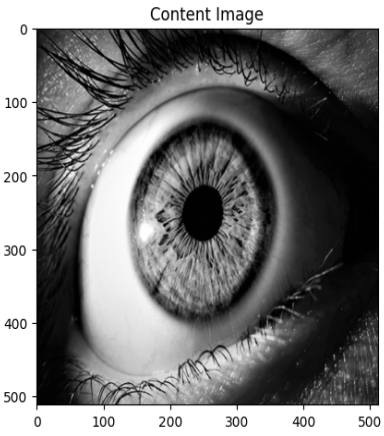

Output Image

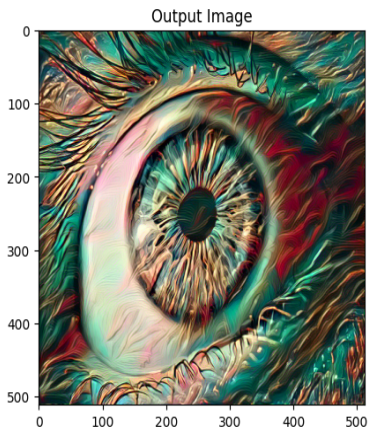

Style Image 1


Content Image


Output Image


Style Image 1


Content Image


Output Image

Output Visualization

- The resulting images blend the structural integrity of the content image with the stylistic patterns and textures of the style images.
- Visual inspection shows successful integration of styles without significant loss of content details.

Performances and Challenges

Strengths:

- The model demonstrates effective transfer of multiple artistic styles onto the content image.
- Pretrained weights reduce the computational load, improving convergence.

Challenges:

- High computational cost for larger images and multiple style sources.
- Style blending weights may require manual tuning for specific aesthetic goals.

Applications

Neural Style Transfer finds practical use in:

- **Art Creation:** Transforming photographs into visually appealing artworks.
- **Augmented Reality (AR):** Applying artistic effects in real time for AR experiences.

- *Video Stylization:* Extending the concept to stylize video frames while maintaining temporal coherence.

Future Steps

Memory Optimization:

- Utilize lower-resolution inputs or compress feature representations for faster computations.

Dynamic Weight Adjustments:

- Introduce adaptive weighting to balance style contributions from multiple images dynamically.

Integration with GANs:

- Explore Generative Adversarial Networks for real-time style transfer and improved image quality.

Interactive User Interface:

- Develop an interface allowing users to control and fine-tune style blending parameters interactively.

Beyond Images:

- Research into applying NST to other domains, such as text and music, is ongoing.

References

- Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A Neural Algorithm of Artistic Style. arXiv:1508.06576.
- Ferlatti, A. (2023). Neural Style Transfer: Theory and Implementation. Medium.
- Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. arXiv:1603.08155.
- Huang, X., & Belongie, S. (2017). Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization (AdaIN). arXiv:1703.06868.

The link to the colab notebook is given here.