Style Transfer Using Convolutional Neural Networks

CSCI 611 – Spring 2025
Author: Tirth Bhavesh Shah

---

1. Introduction

Style transfer is a deep learning technique that applies the artistic style of one image to the content of another using convolutional neural networks (CNNs). This report details the implementation of the style transfer method introduced by **Gatys et al. (2016)** using a pre-trained **VGG19** network in PyTorch. The goal is to blend the style of one image (e.g., a painting) with the content of another (e.g., a photograph), producing an aesthetically appealing output.

2. Methodology

2.1 Pretrained VGG19 for Feature Extraction

A pre-trained VGG19 network was used to extract image features. Lower layers capture textures and edges, while deeper layers encode more abstract content representations.

The layers used for feature extraction include:

- Content Layer: conv4_2
- Style Layers: conv1_1, conv2_1, conv3_1, conv4_1, conv5_1

2.2 Gram Matrix for Style Representation

The style representation of an image was obtained using the Gram Matrix, which measures correlations between different feature maps. This provides a strong sense of texture and artistic style.

2.3 Loss Functions

The total loss function consists of two components:

- Content Loss: Preserves the structure and content of the original image.
- Style Loss: Encourages the target image to match the artistic style of the style image.

2.4 Optimization Process

We initialized the target image as a copy of the content image and updated it iteratively using **Adam or LBFGS optimizers**. The learning rate and step size were adjusted to balance style and content representation.

3. Experiments and Hyperparameter Tuning

We experimented with multiple hyperparameter settings to observe their impact on results.

3.1 Number of Iterations

- **500 iterations:** Quick results, but the style wasn't well-transferred.
- **1000 iterations:** Better details and smoother textures.
- **2000 iterations:** High-quality transfer but computationally expensive.

3.2 Content vs. Style Weight Ratio

Content Weight (α\alphaα)	**Style Weight (β\betaβ)**	**Output Effect**
1e4	1e6	High emphasis on artistic style, less content structure
1e5	1e5	Balanced mix of content and style
1e5	1e4	More content preservation, less stylization

3.3 Learning Rate

- **0.005:** Converged fast but produced unstable artifacts.
- **0.003:** Provided smoother results with stable convergence.
- **0.001:** Slower convergence but consistent results.

4. Results and Discussion

- Higher **style weights** resulted in better stylization but sometimes distorted content.
- Lower **content weights** resulted in more abstract, painterly effects.
- **Reducing image size** to **256x256** significantly sped up processing without major quality loss.
- **Using LBFGS optimizer** instead of Adam led to faster convergence on a CPU.

5. Conclusion

This project successfully implemented **neural style transfer** using **VGG19** and explored the impact of hyperparameter tuning. Key findings include:

- Balancing **content and style weights** is crucial for achieving realistic results.
- **Optimizers and image size** significantly affect processing speed.