Style Transfer with Neuro Network

Yan Lin Htet
Department of Industrial Systems Engineering (MMI)
Asian Institute of Technology
Pathum Thani, Thailand
st125167@ait.asia

Soe Htet Naing
Department of Information and Communication Technology (DSAI)
Asian Institute of Technology
Pathum Thani, Thailand
st125166@ait.asia

Abstract

Style transfer is a technique in computer vision that combines the visual style of one image with the content of another. This process creates new images that preserve the structural elements of the content image while adopting the aesthetic features, such as colors, textures, and patterns, of the style image.

I. INTRODUCTION

Style transfer, a prominent area of study within computer vision and deep learning, focuses on the transformation of an image by re-imagining it in the visual style of another. The process synthesizes two primary elements: the "content" of one image, which includes its structural and semantic information, and the "style" of another, often reflecting artistic techniques such as brush strokes, colors, and textures. Initially popularized by the artistic results seen in "neural style transfer" techniques, the field has grown significantly, influencing both art and practical applications in digital media.

To overcome these limitations, researchers have explored alternative architectures, including **transformer networks**, which excel in capturing long-range dependencies using self-attention mechanisms. These advancements have paved the way for more efficient and high-quality style transfer methods. This paper reviews the methodologies and progress in style transfer using VGG16 and transformers, highlighting their strengths, limitations, and potential future directions.

II. Literature Review

Style transfer is a technique for re-rendering images by blending content and artistic styles, pioneered by **Gatys et al. (2015)** using CNNs like VGG16. VGG16-based methods extract hierarchical features from content and style images, leveraging the Gram matrix for style representation. While effective, these methods are computationally intensive and require iterative optimization. To address efficiency, fast neural style transfer (Johnson et al., 2016) introduced feed-forward networks for real-time processing.

More recently, **transformer networks** have emerged as powerful alternatives. Their self-attention mechanisms capture global dependencies, enabling superior performance in complex style transfers. Vision Transformers (ViTs) and diffusion models push the boundaries of style blending and quality. However, both approaches face challenges in computational efficiency, multi-style transfer, and user control. Future research aims to hybridize CNNs and transformers, enhance realism with GANs, and extend style transfer to other modalities.

III. Methodology

The methodology of this research focuses on implementing neural style transfer (NST) to transform the artistic style of an image onto a target image while preserving the content structure of the latter. This was achieved using a deep learning approach inspired by the VGG-16 architecture, widely regarded for its ability to extract meaningful hierarchical features. The process involved designing and training a convolutional neural network (CNN)-based Transformer Net to balance style and content representation through optimization techniques.

i. Dataset Preparation

The dataset for training consisted of unlabeled images sourced from the COCO dataset (test2017 subset), which was downloaded and preprocessed. Images were resized to 256 × 256 dimensions and normalized using mean and standard deviation values to ensure consistent input for the neural network. The style image, selected for its distinctive artistic features, underwent similar preprocessing to enable its integration into the style transfer process.

ii. Model Architecture

Two primary networks were utilized:

1. **VGG16 Feature Extraction Network**

A modified VGG16 model serves as the backbone for extracting multi-level image features critical to the style transfer process.

The network is divided into slices corresponding to its convolutional layers: relu1_2, relu2_2, relu3_3, and relu4_3. These layers capture hierarchical features ranging from low-level textures to high-level representations.

During training, the VGG16 model remains frozen to ensure computational efficiency and focus on style transfer. The extracted features are later used to compute content loss and style loss.

2. **Transformer Network Architecture**

The transformer network consists of a fully convolutional design capable of processing entire images in a single forward pass. It is designed with three main components:

Down sampling Layers: These layers use stride convolution to reduce spatial dimensions while increasing feature depth.

Residual Blocks: A series of five residual blocks enhances the network's capacity to learn complex transformations while maintaining computational stability and avoiding vanishing gradients.

Up sampling Layers: These layers employ transposed convolution to restore spatial resolution, ensuring that the transformed output maintains the same size as the input.

The transformer network is trained to learn a mapping from input content images to stylized outputs.

iii. Loss Functions

Two primary loss functions guide the training process:

Content Loss: Measures the similarity between the feature representations of the content image and the stylized output at the relu2_2 layer of the VGG16 model. This ensures that the structural details and spatial layout of the content image are preserved.

$$L_{content} = \frac{1}{2}\sum\left(F_{ij}^{gen} - F_{ij}^{content}\right)^2$$

Here,F_{ij} are feature activations for the i^{th}layer and j^{th}neuron.

Style Loss: Compares the Gram Matrices of feature maps extracted from the style image and the stylized output. This captures the texture and overall artistic patterns of the style image at multiple layers (relu1_2, relu2_2, relu3_3, and relu4_3).

The Gram matrix is used to describe the correlations between the features in the convolutional layers. It is calculated as:

$$G_{ij} = \sum_{k} F_{ik}F_{jk}$$

Where $\boldsymbol{F}$ is the feature map from a given layer, $\boldsymbol{i}$ and $\boldsymbol{j}$ represent spatial locations, and $\boldsymbol{k}$ represents the feature channels. This matrix encodes the correlations of features and is central to style transfer, as it helps to maintain the textures and patterns of the style image in the generated output.

$$L_{style} = \frac{1}{4N^2M^2}\sum(G^{gen} - G^{style})^2$$

N is the number of filters, and ***M*** is the size of each feature map.

Total loss: is computed as a weighted sum of content and style loss, with hyperparameters $\lambda_{content}$ and λ_{style} controlling their respective contributions.

$$L_{total} = \alpha L_{content} + \beta L_{style}$$

α = content weight, β = style weight

iv. Training Process

The model was trained using the COCO dataset for the content images and a selected style image (e.g., Picasso's self-portrait). The training pipeline includes the following steps:

Data Loading: Content images were loaded in batches and passed through a series of transformations, including resizing, random cropping, and normalization.

Forward Pass: Content images were passed through the transformer network to generate stylized outputs, which were subsequently analyzed by the VGG16 model to extract content and style features.

Loss Computation: The content and style losses were computed using the extracted features, with their gradients back-propagated through the transformer network.

Optimization: The Adam optimizer was used with a learning rate of 1e−5 to minimize the total loss and update the transformer network's parameters.

v. Model Evaluation

The model's performance was evaluated both qualitatively and quantitatively:

Qualitative Assessment: Stylized outputs were generated using sample images, assessing the visual balance between content preservation and style intensity.

Quantitative Metrics: Average content loss and style loss during training were monitored to identify convergence and compare performance across experiments. The model with the lowest total loss (weighted sum of content and style losses) was selected as the best-performing checkpoint for deployment.

vi. Testing and Visualization

Stylized outputs for test images were generated using the best model. A reprocessing function converted tensors into RGB images for visualization.

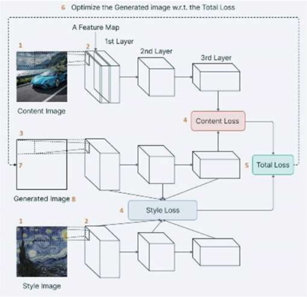

Fig.1: Neural Style Transfer Model Architecture

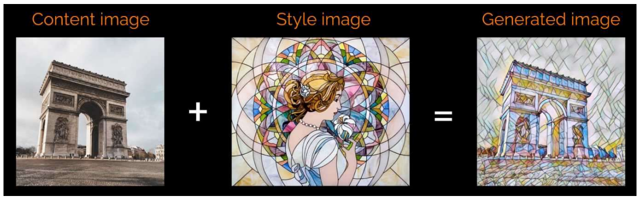

Fig.2: the content and style representations of an image using a deep learning model

IV. Experimental Results

This section summarizes the results obtained from the neural style transfer experiments conducted with different configurations. The primary objective was to evaluate the impact of varying hyperparameters, including the number of epochs, batch size, and loss weights, on the quality of stylized outputs and training efficiency.

1. Experiment Configurations

Three experiments were performed with the following configurations:

Experiment1: Low epochs (2), small batch size (4), and high content weight ($\lambda_{content} = 10^6, \lambda_{style} = 10^9$).

Experiment2: Medium epochs (4), medium batch size (8), and balanced content weight ($\lambda_{content} = 10^5, \lambda_{style} = 10^{10}$).

Experiment3: High epochs (8), large batch size (16), and high content weight ($\lambda_{content} = 10^4, \lambda_{style} = 10^{11}$).

2. Qualitative Results

(Exp1): Images maintain strong content structure but show limited style incorporation (Figure 1).

(Exp2): Strikes a balance between content retention and style fidelity, producing the most visually appealing results (Figure 2).

(Exp3): Emphasizes artistic stylization, occasionally at the expense of content details (Figure 3).

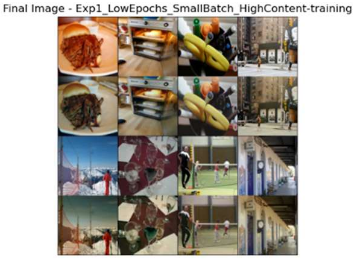
Final Image - Exp1_LowEpochs_SmallBatch_HighContent-training

Figure-1

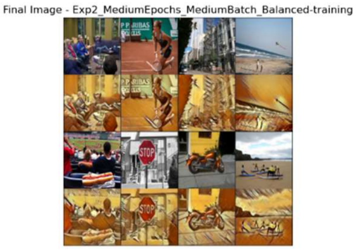
Final Image - Exp2_MediumEpochs_MediumBatch_Balanced-training

Figure-2

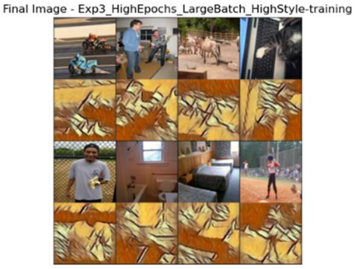

Figure-3

3. Quantitative Analysis

The experiments were evaluated using total loss, the weighted sum of content and style loss:

Experiment 1: Achieved faster convergence due to fewer epochs but exhibited limited stylization (Figure 4).

Experiment 2: Maintained balanced losses during training, leading to smoother and more coherent outputs (Figure 5).

Experiment 3: Slower convergence owing to higher complexity, resulting in the most distinct style representations (Figure 6).

4. Best Model Selection

Visual inspection and quantitative analysis identified **Experiment 2** as the optimal configuration, achieving a harmonious balance between style preservation and content retention.

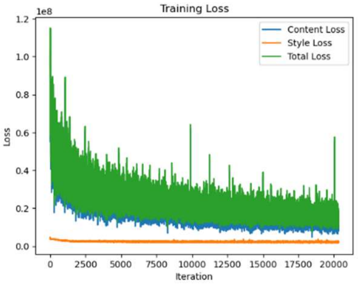

Figure-4

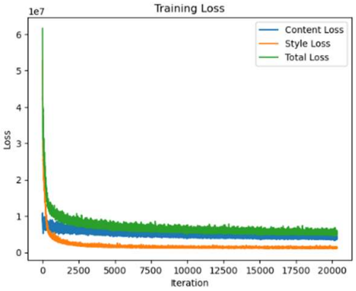

Figure-5

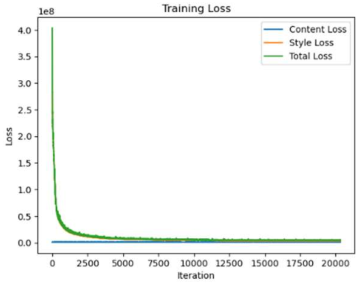

Figure-6

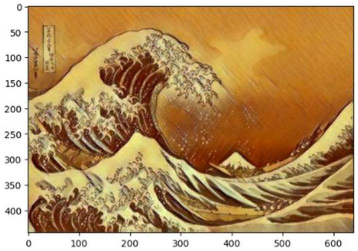

Figure-7

5. Stylized Output

A sample content image processed with the best model demonstrates the blend of artistic textures and realistic content preservation (Figure 7).

6. Observations and Insights

Increasing the style weight $\lambda_{content}, \lambda_{style}$ enhances style fidelity but may overwhelm content details. Extending the number of epochs improves results at the cost of increased computational resources. **Medium configurations (Experiment 2)** provide an efficient trade-off between performance and quality, making them suitable for practical applications.

References

[1] Perceptual Losses for Real-Time Style Transfer and Super-Resolution

[2] Neural Style Transfer: Everything You Need to Know

[3] N.Ashikhmin. Fast texture transfer. IEEE Computer Graphics and Applications, 23(4):38–43, July 2003.

[4] A. Mahendran and A. Vedaldi. Understanding Deep Image Representations by Inverting Them. arXiv:1412.0035 [cs], Nov. 2014. arXiv: 1412.0035.

[5] J. Portilla and E. P. Simoncelli. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. International Journal of Computer Vision, 40(1):49–70, Oct. 2000.