Reproduction and Real-time Implementation of Neural Style Transfer with Interactive Semantic Control

Runyu Gao
2400013132@stu.pku.edu.cn

Jinyang Yan
1311568268@qq.com

Yanjia Dong
2400013169@stu.pku.edu.cn

Abstract

Neural Style Transfer (NST) has garnered significant attention at the intersection of computer vision and artistic creation. This paper aims to reproduce and compare representative style transfer algorithms while proposing a novel interactive framework. We first reproduced the classic optimization-based method by Gatys et al. and the Laplacian-Steered LapStyle method for high-resolution structural preservation. While optimization methods produce high-quality textures, they suffer from slow inference speeds. To address this, we implemented a feed-forward Fast Neural Style Transfer network based on existing perceptual loss frameworks. Furthermore, bridging the gap between global stylization and fine-grained control, we proposed a **Real-time Interactive Semantic Style Transfer** system. By integrating the Segment Anything Model (SAM) with an optimized rendering pipeline, our system allows users to apply distinct styles to specific objects and background regions with millisecond-level latency. Experimental results demonstrate that our system achieves high-quality local stylization while maintaining real-time responsiveness on high-resolution images.

1 Introduction

Image style transfer refers to the process of combining the semantic content of one image with the artistic style of another. Since Gatys et al. first proposed the Neural Style Transfer algorithm based on Convolutional Neural Networks (CNN), the field has developed rapidly. However, the original optimization-based method is computationally expensive and struggles with high-resolution details. Subsequent research, such as Laplacian-Steered NST, introduced structural priors to enhance edge preservation, while Johnson et al. proposed feed-forward networks for real-time applications.

The main contributions of this paper are:

1. **Algorithm Reproduction**: We systematically reproduced Gatys' optimization framework and the Laplacian-Steered progressive optimization strategy.

2. **Efficient Implementation**: We implemented a feed-forward TransformerNet to achieve real-time inference (0.04s per image), facilitating practical application.

3. **Interactive Semantic System**: We developed a novel interactive tool combining SAM and the FastStyle framework. By implementing algorithms for background gap filling and ROI-based rendering, we achieved seamless multi-object stylization.

Preprint. Under review.

2 Methodology

2.1 Reproduction of Gatys' Method

We utilize the VGG-19 network pre-trained on ImageNet as the feature extractor. Our implementation optimizes a target image $\vec{x}$ by minimizing the total loss function $\mathcal{L}_{total}(\vec{x}) = \alpha\mathcal{L}_{content} + \beta\mathcal{L}_{style} + \gamma\mathcal{L}_{tv}$.

1. Content Loss: We use the feature maps F^l and P^l from layer `conv4_2`. The loss is defined as:

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

2. Style Loss: Style is represented by the Gram matrix $G^l \in \mathbb{R}^{N_l \times N_l}$, where G_{ij}^l is the inner product between the vectorized feature maps i and j in layer l:

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \tag{2}$$

The style loss $\mathcal{L}_{style}$ is the MSE between the Gram matrices of the generated image G^l and style image A^l across layers $\{$`conv1_1` ... `conv5_1`$\}$:

$$\mathcal{L}_{style} = \sum_l w_l \frac{1}{4N_l^2 M_l^2}\sum_{i,j}(G_{ij}^l - A_{ij}^l)^2 \tag{3}$$

3. Total Variation (TV) Loss: To ensure spatial continuity and suppress checkerboard artifacts, we implement a TV regularizer:

$$\mathcal{L}_{tv}(\vec{x}) = \frac{1}{2HW}\sum_{i,j}(|x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}|) \tag{4}$$

2.2 Reproduction of LapStyle (Laplacian-Steered NST)

To address structural smudging at high resolutions (1024×1024), we reproduced LapStyle using a progressive optimization strategy with a structural constraint.

1. Laplacian Operator: We extract the high-frequency structural residuals of an image I using a fixed 3×3 Laplacian kernel K:

$$\Delta I = I * K, \quad K = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{5}$$

2. Laplacian Loss: During the Revision Stage, the generated image is constrained to preserve the content's edges:

$$\mathcal{L}_{lap} = \frac{1}{CHW}\sum_{c,i,j}(\Delta I_{gen}^{c,i,j} - \Delta I_{content}^{c,i,j})^2 \tag{6}$$

By combining $\mathcal{L}_{lap}$ with the standard NST objectives, the optimization is "steered" to maintain sharp contours of traditional architecture (e.g., the PKU West Gate) while allowing artistic textures to bloom in smooth regions.

2.3 Implementation of Fast Neural Style Transfer

To overcome the efficiency bottleneck of optimization-based methods, we implemented a feed-forward generator network (`TransformerNet`) based on the architecture proposed by Johnson et al., trained on the COCO2017 dataset.

1. Network Architecture: The generator follows an Encoder-Bottleneck-Decoder architecture utilizing Instance Normalization (IN) to enable style-agnostic feature statistics:

- **Encoder**: Three convolutional layers (kernel sizes $9 \times 9, 3 \times 3, 3 \times 3$) progressively downsample the image to extract spatial features.
- **Bottleneck**: Five Residual Blocks (`ResBlock`) perform deep feature transformations while preserving semantic information through skip connections.
- **Decoder**: Two Transposed Convolutional layers (`ConvTranspose2d`) upsampling the feature maps, followed by a final convolution to reconstruct the RGB image.

2. Perceptual Loss: Instead of per-pixel loss, we train the network using a pre-trained VGG-16 loss network ϕ. The total loss is a weighted sum of feature reconstruction loss (content) and Gram matrix matching loss (style):

$$\mathcal{L}_{total} = \lambda_c ||\phi_j(y) - \phi_j(y_c)||_2^2 + \lambda_s ||G(\phi_j(y)) - G(\phi_j(y_s))||_2^2 \tag{7}$$

2.4 Interactive Semantic Multi-Object Style Transfer

We propose a novel system that integrates the Segment Anything Model (SAM) with the implemented FastStyle models to allow real-time, click-based style switching for individual objects.

1. Semantic Segmentation via ViT-based SAM: In our implementation, semantic object extraction is powered by the Segment Anything Model (SAM). The core of SAM is a heavy **Vision Transformer (ViT-H)** based image encoder.

- **ViT Image Encoder**: The input image is first divided into fixed-size patches, which are then processed through multiple layers of Transformer blocks to extract high-level semantic embeddings.
- **Zero-Shot Mask Generation**: Utilizing the pre-computed embeddings, our system employs the `SamAutomaticMaskGenerator` to perform zero-shot segmentation. By distributing a grid of point prompts over the image, the lightweight mask decoder generates multiple binary masks.

2. Pre-computation and Space-Time Tradeoff: To achieve instant feedback, we moved heavy computations to the loading phase. We pre-compute stylized versions of the full image for all K available styles and pre-calculate Gaussian-blurred Alpha masks α_i for all N objects.

3. ROI-based Rendering Engine: We implemented a Region-of-Interest (ROI) rendering pipeline. When a user toggles the style of an object, the system does not re-render the entire image. Instead, it identifies the object's bounding box and updates only the affected pixels in the display buffer using Alpha Blending:

$$I_{out} = I_{style} \cdot \alpha + I_{base} \cdot (1 - \alpha) \tag{8}$$

This optimization reduces the computational complexity from $O(H \times W)$ to $O(h_{bbox} \times w_{bbox})$, enabling smooth interaction even on $4K$ resolution monitors.

3 Experiments

3.1 Experimental Settings

Content images were collected from **Peking University**. We trained FastStyle models on styles including "Candy", "Feathers", "Starry Night", and "The Great Wave".

3.2 Comparative Analysis: Gatys vs. LapStyle

We compared the generation quality of Gatys' method and LapStyle. Figure 1 shows the result on the PKU West Gate stone lion with the "Candy" style. LapStyle generates clearer geometric textures and better preserves the lion's structure compared to Gatys' method. Similarly, in Figure 2, LapStyle successfully synthesizes the distinct brushstrokes of "Starry Night".

Figure 1: Comparison on the West Gate Stone Lion ("Candy" style). LapStyle (bottom right) produces sharper textures than Gatys (top right).

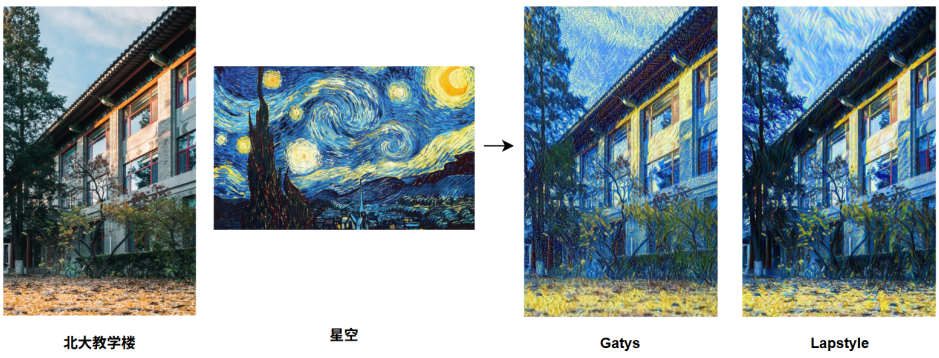

Figure 2: Comparison on campus scenery ("Starry Night" style). LapStyle exhibits more natural artistic brushstrokes.

3.3 Fast Neural Style Transfer Results

The implemented FastStyle models achieve stylization speeds suitable for real-time video streams. As shown in Figure 3, the model successfully transfers global textures while preserving the structural layout of the PKU teaching buildings.

Figure 3: Results of Fast Neural Style Transfer implementation. From left to right: Original Content, feathers, wave and starry night styles.

3.4 Interactive System Evaluation

We evaluated our Interactive Semantic Style Transfer system. Figure 4 demonstrates the user interface where specific regions (e.g., windows, walls, sky) can be individually styled. The system ensures high interactivity by pre-computing object-specific style caches. The interactive demo video is available in the supplementary materials.

Figure 4: Several examples of the Interactive Semantic Style Transfer System.

3.5 Performance Analysis

Table 1 compares the inference time of different methods. Our FastStyle implementation is three orders of magnitude faster than Gatys. Table 2 further highlights the efficiency of our ROI-based rendering strategy in the interactive system.

Table 1: Comparison of Full-Image Inference Time (512×512)

Method	Type	Time (s)
Gatys et al.	Optimization	~ 15.0
LapStyle	Progressive Optimization	~ 25.0
FastStyle (Implemented)	**Feed-forward**	$\sim$ **0.04**

Table 2: Interaction Latency in Semantic System (1200×800 Resolution)

Rendering Strategy	Average Latency (ms)
Global Re-rendering	~ 15.0
ROI-based Partial Update (Ours)	$\sim$ **1.5**

4 Conclusion

We presented a comprehensive reproduction of NST algorithms, from the foundational Gatys method to the structure-preserving LapStyle. To enable practical applications, we implemented a high-efficiency FastStyle network based on perceptual loss. Finally, we introduced a novel Interactive Semantic Style Transfer system. By combining SAM's ViT-based segmentation capabilities with our optimized ROI rendering engine, we achieved a real-time, highly granular creative tool that allows users to reimagine specific parts of an image with distinct artistic styles.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016.

[2] S. Liang, L. Lin, W. Yang, P. Luo, and J. T. Kwok. Laplacian-Steered Neural Style Transfer. In *ACM Multimedia*, 2017.

[3] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[4] A. Kirillov, E. Mintun, N. Ravi, et al. Segment Anything. In *ICCV*, 2023.