Custom Artistic Neural Style Transfer

Project presentation DA 312

Onkar shinde
220150021

Introduction

- **Neural Style Transfer (NST)** is an innovative technique that allows for the fusion of artistic styles from one image with the inherent content of another to generate a unique piece of art.
- **Mechanism of NST:** The process involves the manipulation of pixel values guided by the stylization features defined by deep learning models, particularly to achieve aesthetically pleasing results.
- **Technical Framework:** The project builds upon existing frameworks in PyTorch and leverages the VGG-19 architecture for image feature extraction and processing.
- **Web Interface Features:** A user-friendly Flask web interface facilitates interaction with the NST model, making it accessible for real-time stylization tasks.

Objectives

Artistic stylization using CNN-based feature blending. We leverage deep neural networks for extracting content and style from images, and optimize results for high-quality artistic transformations. The implementation also focuses on real-time delivery and aesthetic post-processing.

01. Artistic stylization via CNN-based feature blending.

02. Use VGG-19 for content/style extraction.

03. Optimization using Adam for efficient convergence.

04. Real-time image stylization through web interface.

05. Post-processing: Color preservation & histogram matching.

Methodology

01

Feature Extraction

Employing the VGG-19 architecture to capture distinct features across various network layers, ensuring a comprehensive grasp of both content and style representations.

02

Content Layer Utilization

Focusing on the conv4_2 layer for its high-quality content representation, maximizing structural fidelity during stylization.

03

Style Layer Utilization

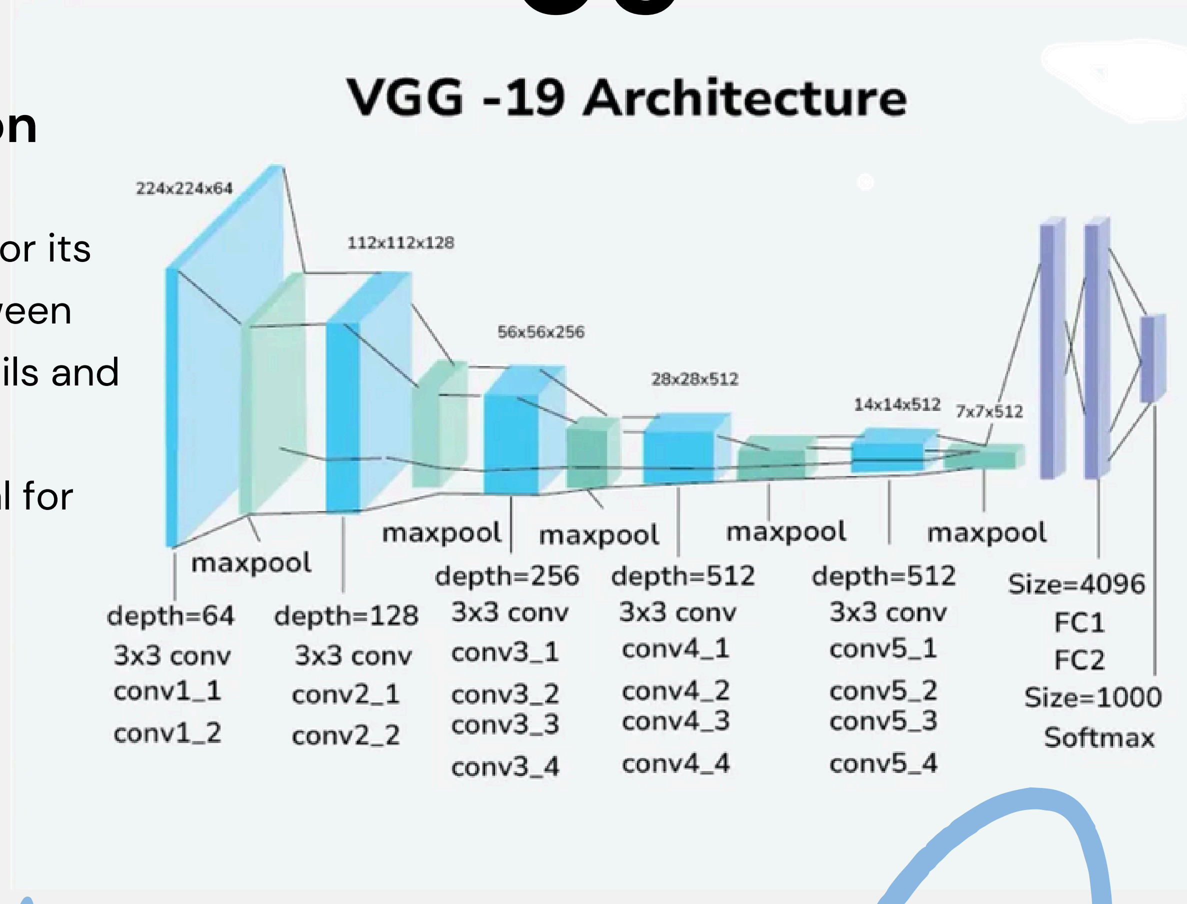
Using lower layers like conv1_1 and conv2_1, which effectively capture textures, helps embed nuanced stylistic elements into the output.

04

Layer Separation

VGG-19 was chosen for its clear separation between lower-level style details and high-level content abstraction—essential for style transfer tasks.

05



Loss Functions for Optimization

📶 Total Loss Function

The total loss is a weighted sum of content, style, and total variation losses.

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

🔣 Content Loss

Measures the difference in content features using Euclidean distance.

$$L_{\text{content}} = \frac{1}{2} \sum_{i,j} (F_{ij}^C - F_{ij}^G)^2$$

🔷 Style Loss

Compares texture similarity using Gram matrices of feature maps.

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

$$L_{\text{style}} = \sum_l w_l \cdot \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

🛡️ Total Variation Loss

Applies a smoothness constraint to reduce noise.

$$L_{\text{TV}} = \sum_{i,j} (x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2$$

System Workflow

- **Image Upload Process:** Users initiate the process by uploading their chosen content and style images, marking the beginning of the NST workflow.
- **Image Preprocessing Steps**: The uploaded images undergo preprocessing to ensure they are ready for feature extraction and subsequent processing.
- **Feature Extraction Through VGG-19:** Following preprocessing, the system invokes the VGG-19 architecture to extract the necessary features for both style and content images.
- **Loss Computation:** After feature extraction, various loss computations take place to evaluate how well the output matches the desired content and style attributes.
- **Optimization Phase:** Using Adam model optimizes the output image based on computed losses to arrive at a final stylized output.
- **Final Image Generation:** The last phase involves generating the final image which integrates both user-generated content and stylistic elements, providing a stylized output.

Web Interface for User Interaction

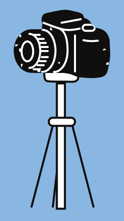
Web Development Frameworks

The front end is constructed utilizing Flask combined with HTML, CSS, and JavaScript, creating a responsive web experience for users.

User Upload Section

Users can upload content and style images directly, enabling personalized neural style transfer outputs.

Input Control



A slider allows users to select how many stylized images they want to generate, offering flexible control.

Progress Indicators

Progress bars and live logs provide real-time updates during processing, keeping users informed throughout.

Image Preview Features

Users can preview and download stylized results, ensuring easy access and a seamless experience.

Output Management

Generated images are organized and displayed in an accessible gallery format, simplifying navigation and comparison.

Webpage Interface

Neural Style Transfer

Neural Style Transfer

Upload Content Image:

Choose File 14.jpg

Upload Style Image:

Choose File blue_swirls.jpg

Number of Output Images (1–10):

2

Apply Style

Applying style, please wait...

Webpage Interface

Terminal Output

```
2025-04-20 11:45:48.404290: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Model loaded.
Starting iteration 1 of 2 (step 0 / 200)
Current loss value: 2512363500.0  Improvement : 0.000 %
Rescaling Image to (400, 711)
Image saved as static/results/23fa451a-2b89-4681-8c91-c7cd20cf84d9_at_iteration_1.png
Iteration 1 completed in 173s
Starting iteration 2 of 2 (step 100 / 200)
Current loss value: 1118890800.0  Improvement : 55.465 %
Rescaling Image to (400, 711)
Image saved as static/results/23fa451a-2b89-4681-8c91-c7cd20cf84d9_at_iteration_2.png
Iteration 2 completed in 174s
```

Generated Images

Download

Download

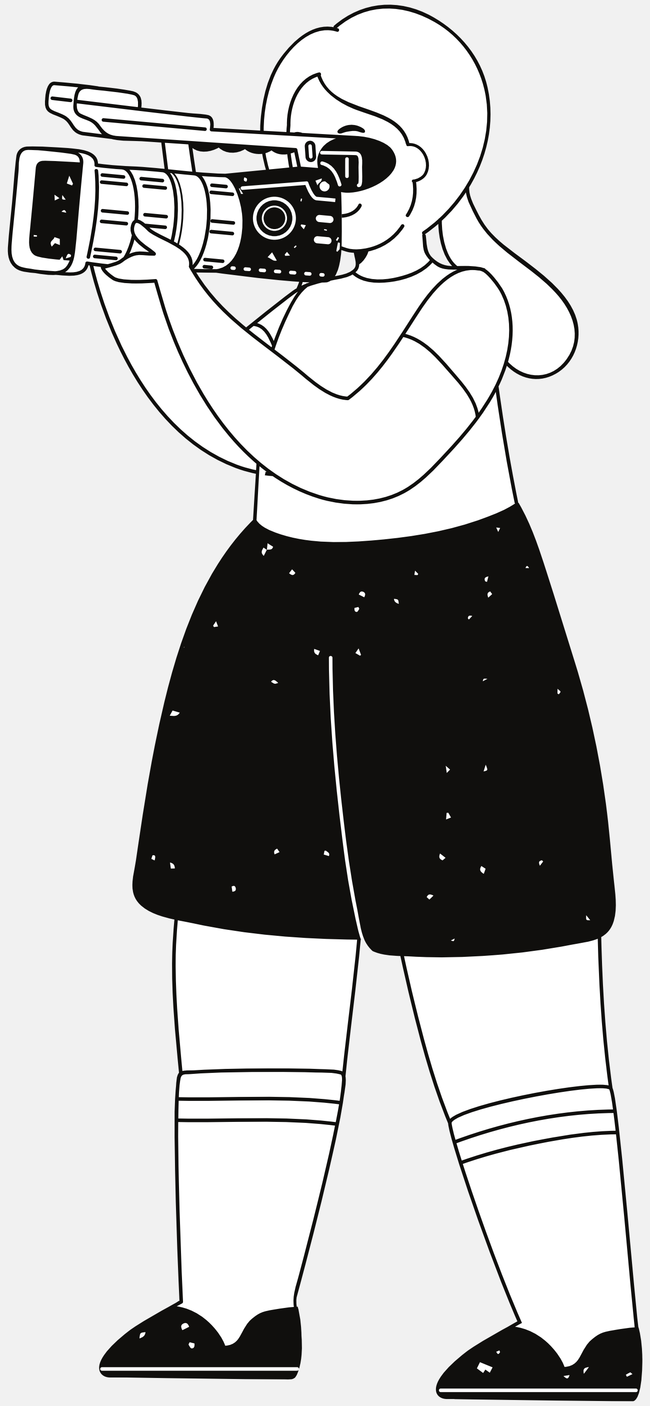
Stylized Results

Content image

Style image

Output image

Output with color correction



Challenges & Solutions

- High compute cost → Resize & optimize images
- Color distortion → Histogram matching, color preservation
- Long iteration time → Adaptive learning rate + user-tuned iterations

Conclusion

- Developed a custom end-to-end Neural Style Transfer (NST) system
- Enabled real-time, interactive, and visually engaging user outputs
- Integrated concepts from CNNs, optimization algorithms, and user interface design
- Provided a complete web-based solution for style transformation
- Demonstrated practical applications of deep learning in creative media

Thank you very much!