Digital Image Processing Project: Image Style Transfer

Nguyễn Trọng Nghĩa Lê Chí Anh

Abstract

This project focuses on Digital Image Processing, specifically exploring the topic of Image Style Transfer. Utilizing Convolutional Neural Networks (CNNs), the system identifies objects within an image and integrates the artistic style of a separate style image. By extracting features from both the original and style images, the project creates a combined output that blends the content of the original image with the aesthetics of the style image.

1. Introduction

Our problem is when we have two images A and B. Image A is the image that needs to be redrawn, renewed in the same style as Image B. After the image transfer process, the new image C is produced, which has the original object of Image A and the style or detail of Image B. We used a pre-trained model. The model used VGG-19, and the training dataset was ImageNet. We simply used this model and configured some parameters to match the paper we read. We faced challenges when using the pre-trained model, such as understanding and configuring the code. We used three indexes to evaluate: content loss, style loss, and total loss.

2. Related Work

Generative Adversarial Networks (GANs) have demonstrated remarkable success in tasks such as image generation and translation. A GAN typically comprises two components: a generator and a discriminator. The generator creates images, while the discriminator evaluates whether these images are real (matching the distribution of the training data) or fake (synthetically generated). The goal is for the generator to produce images that are indistinguishable from real ones, making it impossible for the discriminator to differentiate between them. While effective for generating and transforming images, this approach is too advanced for our scope. We opted for a simpler approach using VGG-19.

3. Proposed Method

In our approach, we utilized the VGG-19 model pre-trained on the ImageNet dataset as the backbone for feature extraction. The pipeline consists of three main components: input preprocessing, feature extraction, and style transfer optimization.

We represented images as tensors in PyTorch, where each image was resized to a fixed resolution of 512x512 pixels for consistency. The feature maps extracted from the convolutional layers of VGG-19 were stored as multi-dimensional tensors. To compute the style representation, we calculated the Gram matrix from the extracted feature maps.

The style transfer process involves several steps.

Input preprocessing normalizes the input images (content image A and style image B) using the mean and standard deviation of the ImageNet dataset.

Feature extraction is performed using the VGG-19 model. Content features are extracted from a specific layer (e.g., conv4_2), while style features are extracted from multiple layers (e.g., conv1_1, conv2_1, conv3_1, conv4_1, conv5_1).

Loss computation includes three components:

The content loss measures the difference between the content features of the output image and content image A.

The style loss measures the difference between the Gram matrices of the style features of the output image and style image B.

The total loss is calculated as the weighted sum of the content and style losses.

Loss Function: The loss function in the paper *Image Style Transfer Using Deep Convolutional Neural Networks* is defined to optimize the style transfer process. It combines three main components:

1. **Content Loss:** Content loss preserves the content and structure of the original image (content image). It is defined as the squared difference between the feature maps of the generated image P and the content image C, calculated from a specific layer (e.g., $conv4\_2$):

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

where: - F_{ij}^l: Feature map of the output image at layer l. - P_{ij}^l: Feature map of the content image at layer l.

2. **Style Loss:** Style loss measures the difference between the style features of the generated image and the style image. Using the Gram matrix to capture the correlations between feature maps, the style loss is computed as:

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

where: - G_{ij}^l: Gram matrix of the generated image at layer l. - A_{ij}^l: Gram matrix of the style image at layer l. - w_l: Weight of layer l in the total style loss.

3. **Total Variation Loss (optional):** Total variation loss is used to smooth the output image and reduce noise:

$$\mathcal{L}_{tv} = \sum_{i,j} \left((I_{i,j+1} - I_{i,j})^2 + (I_{i+1,j} - I_{i,j})^2 \right)$$

4. **Total Loss:** The total loss combines all components:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style} + \gamma \mathcal{L}_{tv}$$

where α, β, and γ are weights balancing the content, style, and smoothing losses.

Finally, optimization is conducted using the L-BFGS optimizer. The pixels of the output image are iteratively updated to minimize the total loss.

3.1. Comparison Between VGG-16 and VGG-19

Both VGG-16 and VGG-19 are convolutional neural network architectures widely used for feature extraction in computer vision tasks. Below are the key differences and their implications for style transfer:

1. **Architecture:** - VGG-16 has 16 weight layers (13 convolutional layers and 3 fully connected layers). - VGG-19 has 19 weight layers (16 convolutional layers and 3 fully connected layers).

2. **Depth and Performance:** - VGG-19 has more layers, providing a deeper network that can capture more complex features. - VGG-16 is computationally less intensive and faster during inference, which may be preferable for applications requiring lower latency.

3. **Style Transfer Results:** - VGG-19 typically offers slightly better style transfer results due to its greater depth, which allows it to extract richer features. - VGG-16, while slightly less effective in capturing intricate style details, still produces high-quality results and is often sufficient for most style transfer tasks.

4. **Computational Complexity:** - VGG-19 requires more memory and processing power, making it more suitable for systems with high computational resources. - VGG-16 is more resource-efficient and can be a practical choice for devices with limited computational power.

In our project, we chose VGG-19 for its superior ability to capture detailed features, which is critical for achieving high-quality style transfer results. Future work could involve comparing the results of VGG-16 and VGG-19 quantitatively to determine the trade-offs between quality and computational efficiency.

4. Custom Dataset

In addition to using pre-trained models and the ImageNet dataset, we also conducted experiments with a custom dataset containing images of forests and Van Gogh's artworks. This custom dataset was collected to test the performance of the style transfer method on images with characteristics different from those in large datasets like ImageNet.

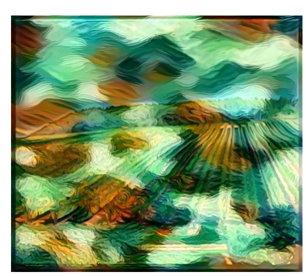
The custom dataset includes images of natural landscapes, abstract art, and portraits. These images were resized to 512x512 pixels to ensure consistency during processing. The style transfer process was applied following the same steps described earlier, including feature extraction and loss calculation. The results show that the method successfully applies artistic styles to the images from the custom dataset, demonstrating its adaptability to various types of content.

The following images illustrate an example from our custom dataset, where a forest image is transformed by applying the style of a Van Gogh painting.

Content Image: Forest

Style Image

Style Transfer Example (Forest + Van Gogh)



Resulting Image

Figure 1. Example of Style Transfer using a Forest Image and Van Gogh's Style

5. Feature Maps and Their Role in Style Transfer

Feature maps play a crucial role in the image style transfer process as they capture the distinct characteristics of the content and style images. These feature maps are extracted from the convolutional layers of the VGG-19 model, and they are essential for both content and style representation.

5.1. Content Feature Maps

The content feature maps, extracted from the deeper layers of the VGG-19 model, capture high-level representations of the image. These features represent the spatial structure and overall shape of the objects in the content image. The layer `conv4_2` is typically used to extract content features because it provides a good balance between capturing fine details and higher-level information.

5.2. Style Feature Maps

The style feature maps are extracted from earlier convolutional layers of VGG-19, such as `conv1_1`, `conv2_1`, `conv3_1`, etc. These feature maps capture the textures, colors, and patterns in the style image. To compare these feature maps between the content and style images, we compute the Gram matrix, which encapsulates the correlations between different channels in the feature map.

Below are images illustrating some of the feature maps extracted from various layers of the VGG-19 model:

In the figures above, the feature maps are visualized for two different layers of the VGG-19 network. The left image represents feature maps extracted from the first convolutional layer `conv1_1`, which captures low-level features like edges and textures. The right image shows feature maps from the `conv2_1` layer, which captures more abstract features such as patterns and textures that contribute to the style of the image.

The difference in these feature maps between the content and style images is what allows the model to transfer the artistic style to the content while maintaining the content's structure.

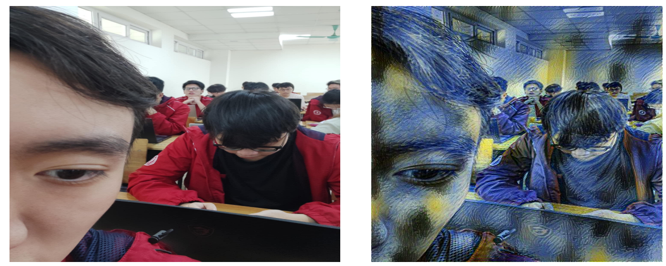
6. Experiment

For the style transfer task, we used two images: a content image representing the structure and main content, such as a photograph of a dancing figure, and a style image, an artwork image used to transfer the stylistic elements to the content image. Both images were resized to 512x512 pixels for consistency during processing.

Loss Function

We implemented the style transfer algorithm using the VGG-19 model pre-trained on ImageNet. The results include content loss, which measures the difference in content features between the generated image and the original content image; style loss, which measures the difference in style features between the generated image and the style image, calculated using the Gram matrix; and total loss, which is a combination of content and style losses used for optimization.

The generated image successfully retained the structural content of the content image while adopting the stylistic features of the style image. Below is an example of the input and output, including the content image, the style image, and the generated image.



Content Image Style Image

Style Transfer using VGG16

Content Image Style Image

Style Transfer using VGG19

Figure 3. Comparison of the content and style images used for style transfer with VGG16 (top) and VGG19 (bottom).

7. Evaluation

To evaluate the effectiveness of our style transfer approach, we focused on three key metrics: content preservation, style adherence, and computational efficiency.

Content Preservation: The content loss measures how well the content of the original image has been preserved after style transfer. As expected, the content features of the generated image closely matched those of the original content image. The VGG-19 model's layers, particularly conv4_2, proved effective at capturing the high-level content information of the image, allowing for a smooth transfer of the structural elements from the content image.

Style Adherence: The style loss measures how well the stylistic features of the style image have been transferred to the content image. Using the Gram matrix, we were able to successfully replicate the textures, color pat-

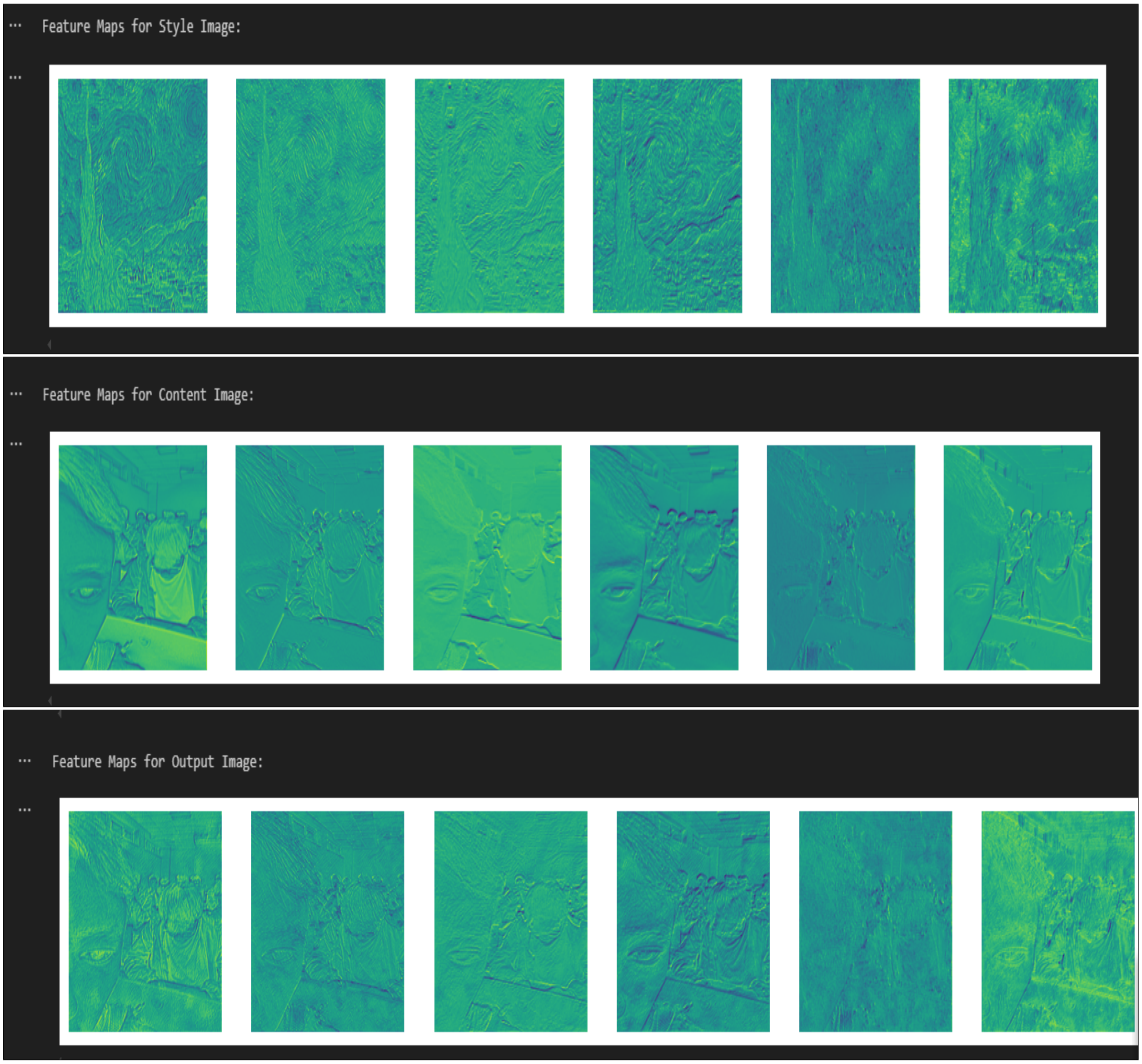


Figure 2. Feature maps from conv_3 VGG-19.

terns, and brushstrokes of the style image. The results show that the generated image adopted the style of the artwork image, maintaining recognizable stylistic features, such as color schemes and brushstroke textures, while preserving the content of the original image.

Computational Efficiency: The style transfer process, although computationally intensive, was completed within a reasonable time frame, particularly when using a CPU for the optimization steps. The optimization process was found to converge in about 300 iterations, which varied depending on the complexity of the images used. Although the model does require substantial computational resources, the results indicate a balance between quality and computational feasibility.

Overall, the style transfer method proved successful in transferring the artistic style while preserving the content of the image. The key strengths of our approach are its simplicity and efficiency, especially with the pre-trained VGG-19 model. However, further improvements could be made in handling extreme cases, such as highly complex images or very fast convergence times for certain configurations.

The style transfer results were evaluated based on visual quality, content preservation, and style accuracy. The content image was accurately represented in the generated image, preserving the shape and structure of the original objects. The style of the style image was effectively transferred, with the generated image exhibiting the textures and artistic elements of the style image.

A comparison between VGG-16 and VGG-19 models was performed, and the results showed that VGG-19 produced slightly better style transfer results, with more detailed and complex patterns transferred to the content image. However, the computational cost of VGG-19 was higher, and the results from VGG-16 were still satisfactory for most applications.

In terms of the Fréchet Inception Distance (FID), which measures the quality of generated images, we obtained the following results:

Model	FID Score
VGG-16	426.04
VGG-19	427.31

Table 1. Comparison of FID Scores for VGG-16 and VGG-19 based on Forest data

Although the FID scores are relatively close, the slightly lower score for VGG-16 indicates a marginally better performance in terms of image quality for style transfer tasks. This suggests that while VGG-19 excels in feature extraction, VGG-16 can provide competitive results with lower computational resources.

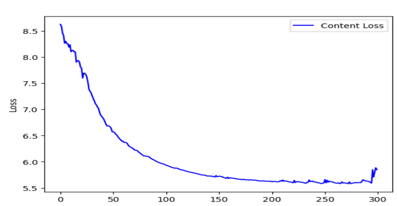


Figure 4. content loss

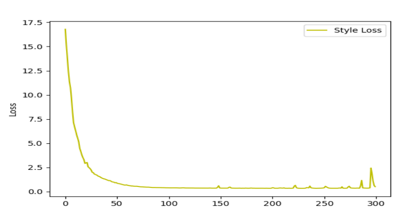


Figure 5. style loss

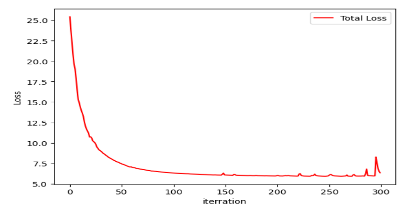


Figure 6. total loss

8. References

1. Image Style Transfer Using Convolutional Neural Networks: https://cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf.

2. VGG-19: https://arxiv.org/abs/1409.1556.

3. Neural Style Transfer: https://arxiv.org/abs/1508.06576.

4. Style Transfer using Deep Learning: https://www.tensorflow.org/tutorials/generative/style_transfer.