

Final Project: Artistic Style Transfer using Deep Learning

Course: *Image Processing and Computer Vision*

Program: *Artificial Intelligence with Machine Learning*

Education Institution: *Humber College*

By Nishit Rathod, Student No N01586439



Workflow of the code.

1. **Importing Libraries:** The code starts by importing several Python libraries required for the Artistic Style Transfer process. These include `torch` (PyTorch) for deep learning, `torchvision` for image transformations and models, `PIL` for handling images, `matplotlib` for visualization, and `os` for file operations.
2. **Mounting Google Drive:** In the next section, the code mounts Google Drive to the Colab notebook. This allows access to files stored in Google Drive.
3. **Loading VGG16 Model:** The code loads the VGG16 model pretrained on the ImageNet dataset. VGG16 is a deep neural network commonly used for image-related tasks.
4. **Freezing Model Parameters:** It sets the `requires_grad` property of all VGG16 parameters to False, which means that these parameters will not be updated during the training process. This is because we only want to use the model for feature extraction and not train it from scratch.
5. **Loading and Preprocessing Images:** The code defines a function (`load_and_preprocess_images`) to load and preprocess images from the specified directories. It loads the content and style images, resizes them, converts them to tensors, normalizes them, and moves them to the GPU (if available) for faster computation.
6. **Defining Content and Style Layers:** The code specifies the layers from VGG16 that will be used to extract content and style features. These layers are selected based on their ability to capture content and style information from images.
7. **Feature Extraction:** Another function (`get_features`) is defined to extract features from the loaded images using the VGG16 model. It runs the images through the model and stores the features from specified layers.
8. **Gram Matrix Calculation:** A function (`gram_matrix`) is provided to calculate the Gram matrix of a given tensor. The Gram matrix is used to represent the style of an image by calculating correlations between different channels in the feature maps.
9. **Denormalizing and Converting Images:** Two functions (`tensor_to_image` and `get_generated_image`) are defined to

convert tensor images to PIL images and vice versa. The `get_generated_image` function initializes the generated image as a copy of the content image, which will be optimized to match the style of the style image.

10. **Setting Content and Style Weights:** Content and style weights are specified to control the importance of content preservation and the strength of style transfer in the final generated image.
11. **Stylization Loop:** The code loops over all combinations of content and style images. For each pair, it extracts content and style features, initializes the generated image, and defines the optimizer for optimization.
12. **Optimization Steps:** The main part of the code is the optimization loop. It runs for a fixed number of steps (`steps`) to minimize the total loss, which is a combination of content and style losses. In each step, the generated image is updated using backpropagation.
13. **Saving and Displaying Output:** After the optimization process, the final stylized image is saved with a unique name based on the content and style image combination. Additionally, the style, content, and generated images are displayed using matplotlib.

In summary, the code performs Artistic Style Transfer using the VGG16 model. It takes a set of content and style images, extracts features from the VGG16 model, and optimizes a generated image to combine the content of the content image with the style of the style image. The result is a stylized output image that reflects the artistic style of the style image while preserving the content of the content image.

Python Libraries

```
In [1]: import torch
import torch.nn as nn
import torch.optim as optim
import torchvision.transforms as transforms
import torchvision.models as models
from PIL import Image
import matplotlib.pyplot as plt
import os
```

Checking GPU, Mounting Google Drive and Setting Directory Path.

```
In [2]: # use GPU
device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
print(device)
```

cuda:0

```
In [3]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
In [4]: %cd /content/drive/MyDrive/Dataset
```

/content/drive/MyDrive/Dataset

Load VGG16 model pretrained on ImageNet

```
In [5]: # Load VGG16 model pretrained on ImageNet
vgg16 = models.vgg16(pretrained=True).features
for param in vgg16.parameters():
    param.requires_grad_(False)

vgg16.to(device)
```

```
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the future, please use 'weights' instead.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:223: UserWarning: Arguments other than a weight enum or 'None' for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equivalent to passing `weights=VGG16_Weights.IMAGENET1K_V1`. You can also use `weights=VGG16_Weights.DEFAULT` to get the most up-to-date weights.
  warnings.warn(msg)
Downloading: "https://download.pytorch.org/models/vgg16-397923af.pth" to /root/.cache/torch/hub/checkpoints/vgg16-397923af.pth
100%|██████████| 528M/528M [00:03<00:00, 169MB/s]
```

```

Out[5]: Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)

    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (18): ReLU(inplace=True)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (20): ReLU(inplace=True)
    (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (22): ReLU(inplace=True)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (25): ReLU(inplace=True)
    (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (27): ReLU(inplace=True)
    (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (29): ReLU(inplace=True)
    (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
)

```

The architecture is represented as a sequential stack of layers. Each layer is denoted by a number in parentheses, and the layer type is mentioned alongside its parameters. Let's break down the architecture:

- **Input Layer:** This neural network takes a 3-channel (RGB) image as input.
- **Convolutional Layers:** There are a total of 13 convolutional layers, represented by `Conv2d`, followed by ReLU activation functions (`ReLU(inplace=True)`). Each `Conv2d` layer performs a 2D convolution operation on the input with a specific kernel size, stride, and padding.
- **MaxPooling Layers:** There are 5 MaxPooling layers, represented by `MaxPool2d`. These layers perform 2D max-pooling on the feature maps, reducing their spatial dimensions while retaining important features.
- **VGG16 Output:** The final output of the VGG16 model is the result of the last MaxPooling layer, which produces a tensor with reduced spatial dimensions.

The number in the parentheses associated with each layer represents the layer's order in the sequential stack. For example, `(0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))` refers to the first convolutional layer. It takes a 3-channel input, performs a 2D convolution with 64 output channels, using a 3x3 kernel, stride of 1, and padding of 1.

Similarly, `(17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))` refers to the 18th convolutional layer. It takes a 256-channel input (the output from the previous layer) and produces a 512-channel output after performing a 2D convolution with a 3x3 kernel, stride of 1, and padding of 1.

The `ReLU(inplace=True)` means that the ReLU activation function is applied element-wise to the output of the corresponding `Conv2d` layer, and the operation is performed in-place, which saves memory.

Overall, VGG16 is a deep CNN model with multiple convolutional and max-pooling layers, which allows it to capture and learn hierarchical representations of images, making it suitable for various computer vision tasks.

Function to load and preprocess a list of images

```

In [6]: # Function to load and preprocess a list of images
def load_and_preprocess_images(image_paths, image_size):
    images = []
    for image_path in image_paths:
        image = Image.open(image_path).convert("RGB")
        transform = transforms.Compose([
            transforms.Resize(image_size),
            transforms.ToTensor(),
            transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
            transforms.Lambda(lambda x: x.unsqueeze(0))
        ])
        images.append(transform(image).to(device))
    return images

# Content and style image directories

```

```
content_image_dir = "ArtisticStyleTransferDataset/Content/"
style_image_dir = "ArtisticStyleTransferDataset/Style/"

# List all image files in the content and style directories
content_image_paths = [os.path.join(content_image_dir, filename) for filename in os.listdir(content_image_dir)]
style_image_paths = [os.path.join(style_image_dir, filename) for filename in os.listdir(style_image_dir)]

# Load and preprocess content and style images
content_images = load_and_preprocess_images(content_image_paths, image_size=(384, 384))
style_images = load_and_preprocess_images(style_image_paths, image_size=(256, 256))
```

Function to extract features from VGG16 and print their names and sizes

```
In [7]: # Function to extract features from VGG16 and print their names and sizes
def print_features(image, model):
    x = image
    for name, layer in enumerate(model):
        x = layer(x)
        print(f"Layer {name}, Name: {layer.__class__.__name__}, Size: {x.size()}")

# Print features for content and style images
print("Content Image Features:")
print_features(content_images[0], vgg16)
print("\nStyle Image Features:")
print_features(style_images[0], vgg16)
```

Content Image Features:

```
Layer 0, Name: Conv2d, Size: torch.Size([1, 64, 384, 384])
Layer 1, Name: ReLU, Size: torch.Size([1, 64, 384, 384])
Layer 2, Name: Conv2d, Size: torch.Size([1, 64, 384, 384])
Layer 3, Name: ReLU, Size: torch.Size([1, 64, 384, 384])
Layer 4, Name: MaxPool2d, Size: torch.Size([1, 64, 192, 192])
Layer 5, Name: Conv2d, Size: torch.Size([1, 128, 192, 192])
Layer 6, Name: ReLU, Size: torch.Size([1, 128, 192, 192])
Layer 7, Name: Conv2d, Size: torch.Size([1, 128, 192, 192])
Layer 8, Name: ReLU, Size: torch.Size([1, 128, 192, 192])
Layer 9, Name: MaxPool2d, Size: torch.Size([1, 128, 96, 96])
Layer 10, Name: Conv2d, Size: torch.Size([1, 256, 96, 96])
Layer 11, Name: ReLU, Size: torch.Size([1, 256, 96, 96])
Layer 12, Name: Conv2d, Size: torch.Size([1, 256, 96, 96])
Layer 13, Name: ReLU, Size: torch.Size([1, 256, 96, 96])
Layer 14, Name: Conv2d, Size: torch.Size([1, 256, 96, 96])
Layer 15, Name: ReLU, Size: torch.Size([1, 256, 96, 96])
Layer 16, Name: MaxPool2d, Size: torch.Size([1, 256, 48, 48])
Layer 17, Name: Conv2d, Size: torch.Size([1, 512, 48, 48])
Layer 18, Name: ReLU, Size: torch.Size([1, 512, 48, 48])
Layer 19, Name: Conv2d, Size: torch.Size([1, 512, 48, 48])
Layer 20, Name: ReLU, Size: torch.Size([1, 512, 48, 48])
Layer 21, Name: Conv2d, Size: torch.Size([1, 512, 48, 48])
Layer 22, Name: ReLU, Size: torch.Size([1, 512, 48, 48])
Layer 23, Name: MaxPool2d, Size: torch.Size([1, 512, 24, 24])
Layer 24, Name: Conv2d, Size: torch.Size([1, 512, 24, 24])
Layer 25, Name: ReLU, Size: torch.Size([1, 512, 24, 24])
Layer 26, Name: Conv2d, Size: torch.Size([1, 512, 24, 24])
Layer 27, Name: ReLU, Size: torch.Size([1, 512, 24, 24])
Layer 28, Name: Conv2d, Size: torch.Size([1, 512, 24, 24])
Layer 29, Name: ReLU, Size: torch.Size([1, 512, 24, 24])
Layer 30, Name: MaxPool2d, Size: torch.Size([1, 512, 12, 12])
```

Style Image Features:

```
Layer 0, Name: Conv2d, Size: torch.Size([1, 64, 256, 256])
Layer 1, Name: ReLU, Size: torch.Size([1, 64, 256, 256])
Layer 2, Name: Conv2d, Size: torch.Size([1, 64, 256, 256])
Layer 3, Name: ReLU, Size: torch.Size([1, 64, 256, 256])
Layer 4, Name: MaxPool2d, Size: torch.Size([1, 64, 128, 128])
Layer 5, Name: Conv2d, Size: torch.Size([1, 128, 128, 128])
Layer 6, Name: ReLU, Size: torch.Size([1, 128, 128, 128])
Layer 7, Name: Conv2d, Size: torch.Size([1, 128, 128, 128])
Layer 8, Name: ReLU, Size: torch.Size([1, 128, 128, 128])
Layer 9, Name: MaxPool2d, Size: torch.Size([1, 128, 64, 64])
Layer 10, Name: Conv2d, Size: torch.Size([1, 256, 64, 64])
Layer 11, Name: ReLU, Size: torch.Size([1, 256, 64, 64])
Layer 12, Name: Conv2d, Size: torch.Size([1, 256, 64, 64])
Layer 13, Name: ReLU, Size: torch.Size([1, 256, 64, 64])
Layer 14, Name: Conv2d, Size: torch.Size([1, 256, 64, 64])
Layer 15, Name: ReLU, Size: torch.Size([1, 256, 64, 64])
Layer 16, Name: MaxPool2d, Size: torch.Size([1, 256, 32, 32])
Layer 17, Name: Conv2d, Size: torch.Size([1, 512, 32, 32])
Layer 18, Name: ReLU, Size: torch.Size([1, 512, 32, 32])
Layer 19, Name: Conv2d, Size: torch.Size([1, 512, 32, 32])
Layer 20, Name: ReLU, Size: torch.Size([1, 512, 32, 32])
Layer 21, Name: Conv2d, Size: torch.Size([1, 512, 32, 32])
Layer 22, Name: ReLU, Size: torch.Size([1, 512, 32, 32])
Layer 23, Name: MaxPool2d, Size: torch.Size([1, 512, 16, 16])
Layer 24, Name: Conv2d, Size: torch.Size([1, 512, 16, 16])
Layer 25, Name: ReLU, Size: torch.Size([1, 512, 16, 16])
Layer 26, Name: Conv2d, Size: torch.Size([1, 512, 16, 16])
Layer 27, Name: ReLU, Size: torch.Size([1, 512, 16, 16])
Layer 28, Name: Conv2d, Size: torch.Size([1, 512, 16, 16])
Layer 29, Name: ReLU, Size: torch.Size([1, 512, 16, 16])
Layer 30, Name: MaxPool2d, Size: torch.Size([1, 512, 8, 8])
```

The above information contains feature information of two sets of images: content image features and style image features. Each set consists of 31 layers, where each layer is represented with its index, name, and size. Let's break down the information for each set:

Content Image Features:

1. Layer 0: Conv2d - Size: torch.Size([1, 64, 384, 384])
2. Layer 1: ReLU - Size: torch.Size([1, 64, 384, 384])
3. Layer 2: Conv2d - Size: torch.Size([1, 64, 384, 384])
4. Layer 3: ReLU - Size: torch.Size([1, 64, 384, 384])
5. Layer 4: MaxPool2d - Size: torch.Size([1, 64, 192, 192])
6. Layer 5: Conv2d - Size: torch.Size([1, 128, 192, 192]) ...
7. Layer 30: MaxPool2d - Size: torch.Size([1, 512, 12, 12])

Style Image Features:

1. Layer 0: Conv2d - Size: torch.Size([1, 64, 256, 256])
2. Layer 1: ReLU - Size: torch.Size([1, 64, 256, 256])
3. Layer 2: Conv2d - Size: torch.Size([1, 64, 256, 256])

4. Layer 3: ReLU - Size: torch.Size([1, 64, 256, 256])
5. Layer 4: MaxPool2d - Size: torch.Size([1, 64, 128, 128])
6. Layer 5: Conv2d - Size: torch.Size([1, 128, 128, 128]) ...
7. Layer 30: MaxPool2d - Size: torch.Size([1, 512, 8, 8])

In both sets, the layers are represented in order from 0 to 30, indicating the sequential stack of layers. Each layer type (e.g., Conv2d, ReLU, MaxPool2d) is followed by the size of the output tensor. The size information shows the dimensions of the feature map for each layer in the format (batch_size, channels, height, width).

For example, the first Conv2d layer in the content image features has an output size of torch.Size([1, 64, 384, 384]). This means that the feature map after this convolutional layer has a batch size of 1, 64 channels, a height of 384 pixels, and a width of 384 pixels.

Similarly, the first Conv2d layer in the style image features has an output size of torch.Size([1, 64, 256, 256]), indicating a feature map with 1 batch, 64 channels, a height of 256 pixels, and a width of 256 pixels.

The given information represents the feature maps extracted from the content and style images at various stages of the VGG16 model. These feature maps will be used in the artistic style transfer process to preserve the content of the content image while transferring the style from the style image to create the final stylized output.

Define content and style layers for feature extraction

Extract features from VGG16 for content and style images

Function to calculate the Gram matrix of a given tensor

Function to denormalize an image tensor and convert it to a PIL image

Initialize the generated image as a copy of the content image

```
In [9]: # Define content and style layers for feature extraction
content_layers = ["29"] # Update this with the correct content layer name
style_layers = ["4", "9", "16", "23", "30"] # Update this with the correct style layer names

# Extract features from VGG16 for content and style images
def get_features(image, model, layers):
    features = {}
    x = image
    for name, layer in enumerate(model):
        x = layer(x)
        if str(name) in layers:
            features[str(name)] = x
    return features

# Function to calculate the Gram matrix of a given tensor
def gram_matrix(tensor):
    _, c, h, w = tensor.size()
    tensor = tensor.view(c, h * w)
    gram = torch.mm(tensor, tensor.t())
    return gram.div(c * h * w)

# Function to denormalize an image tensor and convert it to a PIL image
def tensor_to_image(tensor):
    denormalize = transforms.Normalize(mean=[-2.12, -2.04, -1.80], std=[4.37, 4.46, 4.44])
    image = tensor.squeeze(0).cpu().clone()
    image = denormalize(image).clamp(0, 1)
    return transforms.ToPILImage()(image)

# Initialize the generated image as a copy of the content image
def get_generated_image(content_image):
    return content_image.clone().requires_grad_(True)
```

Artistic Style Transfer Process

In this artistic style transfer process the content loss and style loss are two key components, which combines the content of one image with the style of another image to create a new stylized image. These losses are used to quantify the differences between the generated image and the content image (for content loss) and between the generated image and the style image (for style loss).

1. Content Loss: Content loss measures the difference in content between the generated image and the content image. The goal of the content loss is to ensure that the generated image preserves the essential features and structure of the content image. To calculate the content loss, a feature extraction step is performed on both the content image and the generated image using a pre-trained convolutional neural network (usually VGG16 or VGG19).

The content loss is computed as the mean squared error (MSE) between the feature maps of a specific layer in the network for the content image and the generated image. By minimizing the content loss, the generated image is encouraged to capture the same high-level content details as the original content image. This results in the preservation of the main objects and structures present in the content image.

2. Style Loss: Style loss measures the difference in style between the generated image and the style image. The style of an image refers to its texture, color distribution, and visual patterns. Unlike content loss, which focuses on high-level features, style loss aims to capture the correlations and statistics of features at different layers of the network.

To compute the style loss, the Gram matrix is used on the feature maps extracted from specific layers of the pre-trained neural network for both the style image and the generated image. The Gram matrix represents the correlations between the different feature maps and contains information about the style of the image.

The style loss is calculated as the mean squared error (MSE) between the Gram matrices of the corresponding layers for the style image and the generated image. By minimizing the style loss, the generated image is encouraged to have similar texture, color, and visual patterns as the style image.

The final objective of artistic style transfer is to find a generated image that simultaneously minimizes the content loss and the style loss. This is achieved through an optimization process, where the weights of the content and style losses are adjusted to control the trade-off between preserving the content of the content image and capturing the style of the style image. The optimization process iteratively updates the generated image to strike a balance between content preservation and style transfer, resulting in a visually appealing and harmonious stylized image.

```
In [10]: # Define content weight and style weight for total loss calculation
content_weight = 1 # Adjust this value to control the importance of content preservation
style_weight = 1200 # Adjust this value to control the strength of style transfer

# Create a folder for saving the stylized output images
output_folder = "ArtisticStyleTransferOutput"
os.makedirs(output_folder, exist_ok=True)

# Number of optimization steps
steps = 1000

for content_image, content_path in zip(content_images, content_image_paths):
    for style_image, style_path in zip(style_images, style_image_paths):
        # Extract features from VGG16 for content and style images
        content_features = get_features(content_image, vgg16, content_layers)
        style_features = get_features(style_image, vgg16, style_layers)

        # Initialize the generated image
        generated_image = get_generated_image(content_image)

        # Define the optimizer and set the learning rate
        optimizer = optim.Adam([generated_image], lr=0.01)

        for step in range(steps):
            # Forward pass through VGG16 to extract features from the generated image
            generated_features = get_features(generated_image, vgg16, content_layers + style_layers)

            # Calculate content loss
            content_loss = torch.mean((generated_features[content_layers[0]] - content_features[content_layers[0]]) ** 2)

            # Calculate style loss
            style_loss = 0
            for layer_name in style_layers:
                generated_style = gram_matrix(generated_features[layer_name])
                style_gram = gram_matrix(style_features[layer_name])
                layer_style_loss = torch.mean((generated_style - style_gram) ** 2)
                style_loss += layer_style_loss / len(style_layers)

            # Total loss as a weighted sum of content and style losses
            total_loss = content_weight * content_loss + style_weight * style_loss

            # Backpropagation to update the generated image
            optimizer.zero_grad()
            total_loss.backward()
            optimizer.step()

            # Optional: Print the progress
            if step % 100 == 0:
                print(f"Step {step}/{steps}, Content Loss: {content_loss.item():.4f}, Style Loss: {style_loss.item():.4f}")

        # Convert the final generated image tensor to a PIL image
        generated_image_pil = tensor_to_image(generated_image)

        # Save the stylized output image with a unique name based on content and style image combination
        content_name = os.path.splitext(os.path.basename(content_path))[0]
        style_name = os.path.splitext(os.path.basename(style_path))[0]
        output_filename = f"{content_name}_with_{style_name}.jpg"
        output_path = os.path.join(output_folder, output_filename)
        generated_image_pil.save(output_path)
```

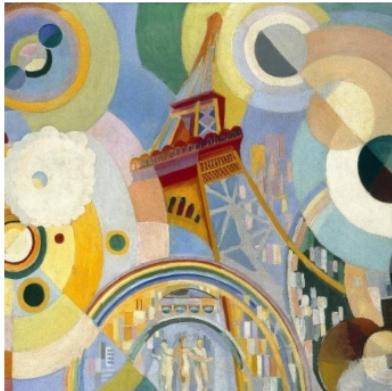
```

# Display the subplots of style, content, and output images
fig, axes = plt.subplots(1, 3, figsize=(15, 5))
axes[0].imshow(Image.open(style_path))
axes[0].set_title("Style Image")
axes[0].axis("off")
axes[1].imshow(Image.open(content_path))
axes[1].set_title("Content Image")
axes[1].axis("off")
axes[2].imshow(generated_image_pil)
axes[2].set_title("Stylized Output")
axes[2].axis("off")
plt.show()

```

Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0004
Step [100/1000], Content Loss: 0.0057, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0041, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0036, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0031, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0029, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0028, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0029, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0026, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0026, Style Loss: 0.0000

Style Image



Stylized Output



Content Image



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0017
Step [100/1000], Content Loss: 0.0294, Style Loss: 0.0003
Step [200/1000], Content Loss: 0.0259, Style Loss: 0.0002
Step [300/1000], Content Loss: 0.0236, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0223, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0212, Style Loss: 0.0001
Step [600/1000], Content Loss: 0.0203, Style Loss: 0.0001
Step [700/1000], Content Loss: 0.0193, Style Loss: 0.0001
Step [800/1000], Content Loss: 0.0187, Style Loss: 0.0001
Step [900/1000], Content Loss: 0.0182, Style Loss: 0.0001

Style Image



Stylized Output



Content Image



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0006
Step [100/1000], Content Loss: 0.0088, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0068, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0061, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0056, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0060, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0053, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0054, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0047, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0044, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



```
Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0011  
Step [100/1000], Content Loss: 0.0187, Style Loss: 0.0001  
Step [200/1000], Content Loss: 0.0136, Style Loss: 0.0000  
Step [300/1000], Content Loss: 0.0119, Style Loss: 0.0000  
Step [400/1000], Content Loss: 0.0104, Style Loss: 0.0000  
Step [500/1000], Content Loss: 0.0097, Style Loss: 0.0000  
Step [600/1000], Content Loss: 0.0093, Style Loss: 0.0000  
Step [700/1000], Content Loss: 0.0089, Style Loss: 0.0000  
Step [800/1000], Content Loss: 0.0085, Style Loss: 0.0000  
Step [900/1000], Content Loss: 0.0083, Style Loss: 0.0000
```

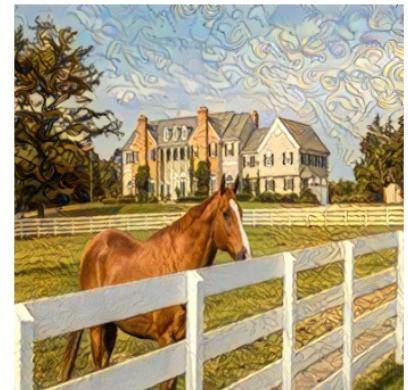
Style Image



Content Image

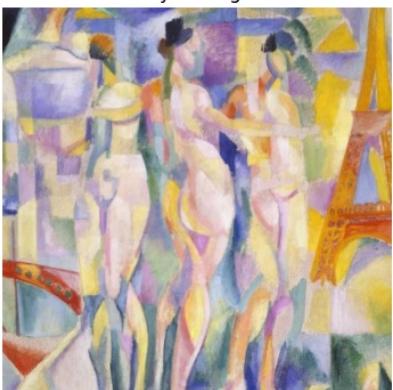


Stylized Output



```
Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0004  
Step [100/1000], Content Loss: 0.0050, Style Loss: 0.0000  
Step [200/1000], Content Loss: 0.0036, Style Loss: 0.0000  
Step [300/1000], Content Loss: 0.0030, Style Loss: 0.0000  
Step [400/1000], Content Loss: 0.0028, Style Loss: 0.0000  
Step [500/1000], Content Loss: 0.0025, Style Loss: 0.0000  
Step [600/1000], Content Loss: 0.0022, Style Loss: 0.0000  
Step [700/1000], Content Loss: 0.0021, Style Loss: 0.0000  
Step [800/1000], Content Loss: 0.0021, Style Loss: 0.0000  
Step [900/1000], Content Loss: 0.0020, Style Loss: 0.0000
```

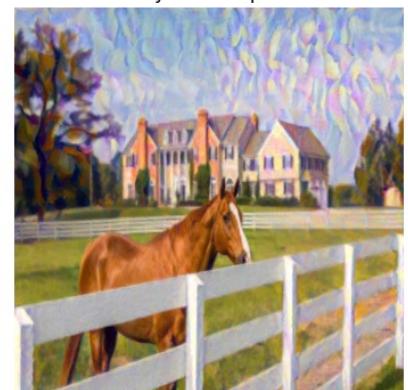
Style Image



Content Image



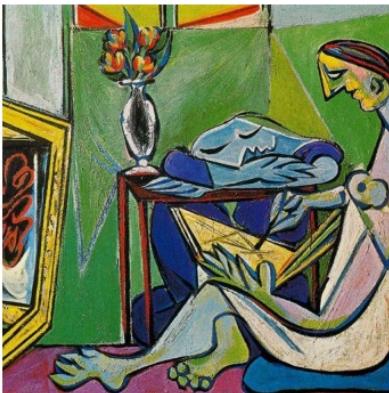
Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0013
Step [100/1000], Content Loss: 0.0125, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0090, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0078, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0068, Style Loss: 0.0000

Step [500/1000], Content Loss: 0.0064, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0059, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0056, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0054, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0052, Style Loss: 0.0000

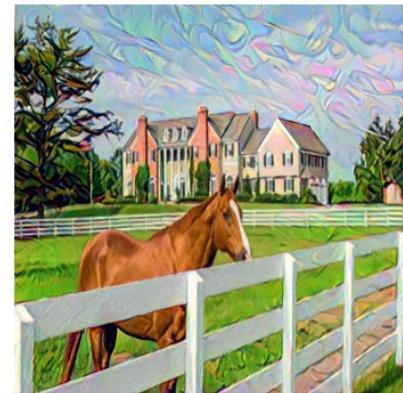
Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0003
Step [100/1000], Content Loss: 0.0049, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0041, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0037, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0030, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0029, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0029, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0029, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0027, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0027, Style Loss: 0.0000

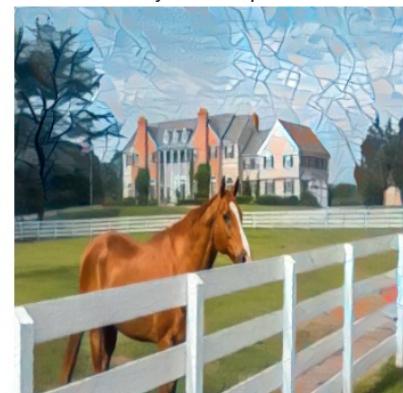
Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0032
Step [100/1000], Content Loss: 0.0406, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0271, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0219, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0190, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0172, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0162, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0150, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0142, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0135, Style Loss: 0.0000

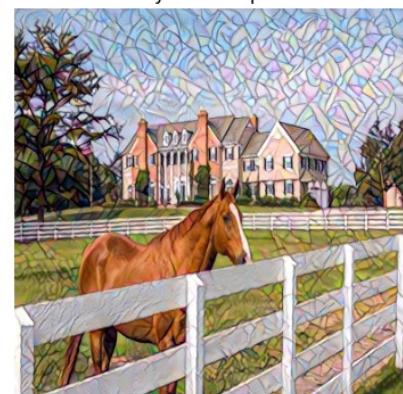
Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0010
Step [100/1000], Content Loss: 0.0101, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0071, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0058, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0055, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0050, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0042, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0039, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0040, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0035, Style Loss: 0.0000

Style Image

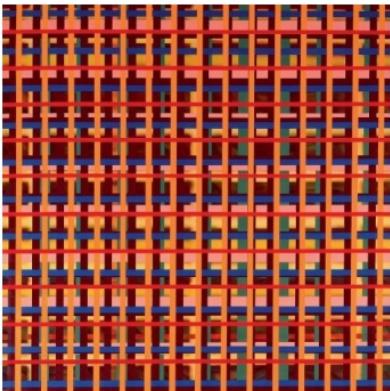


Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0187
Step [100/1000], Content Loss: 0.2191, Style Loss: 0.0023
Step [200/1000], Content Loss: 0.1686, Style Loss: 0.0011
Step [300/1000], Content Loss: 0.1438, Style Loss: 0.0007
Step [400/1000], Content Loss: 0.1281, Style Loss: 0.0005
Step [500/1000], Content Loss: 0.1163, Style Loss: 0.0004
Step [600/1000], Content Loss: 0.1069, Style Loss: 0.0003
Step [700/1000], Content Loss: 0.1001, Style Loss: 0.0003
Step [800/1000], Content Loss: 0.0945, Style Loss: 0.0002
Step [900/1000], Content Loss: 0.0898, Style Loss: 0.0002

Style Image



Stylized Output

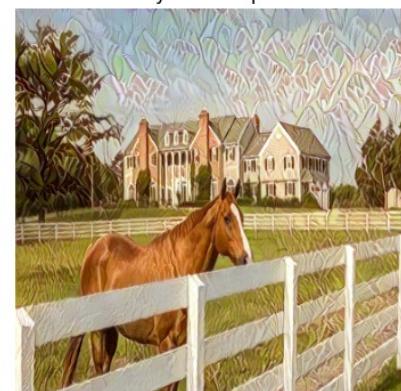


Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0014
Step [100/1000], Content Loss: 0.0210, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0158, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0139, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0128, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0121, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0115, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0111, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0108, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0104, Style Loss: 0.0000

Style Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0009
Step [100/1000], Content Loss: 0.0079, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0055, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0049, Style Loss: 0.0000

Step [400/1000], Content Loss: 0.0038, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0034, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0032, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0030, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0029, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0027, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0009
Step [100/1000], Content Loss: 0.0122, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0092, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0075, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0065, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0060, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0055, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0057, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0051, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0048, Style Loss: 0.0000

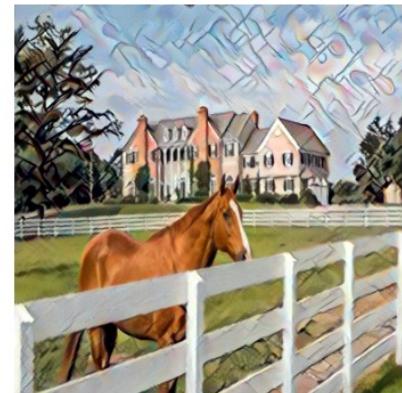
Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0024
Step [100/1000], Content Loss: 0.0254, Style Loss: 0.0003
Step [200/1000], Content Loss: 0.0186, Style Loss: 0.0002
Step [300/1000], Content Loss: 0.0154, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0130, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0121, Style Loss: 0.0001
Step [600/1000], Content Loss: 0.0108, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0100, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0095, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0091, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0027
Step [100/1000], Content Loss: 0.0340, Style Loss: 0.0002
Step [200/1000], Content Loss: 0.0243, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0204, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0180, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0164, Style Loss: 0.0001
Step [600/1000], Content Loss: 0.0151, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0139, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0131, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0125, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0014
Step [100/1000], Content Loss: 0.0203, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0149, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0122, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0106, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0099, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0090, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0086, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0087, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0079, Style Loss: 0.0000

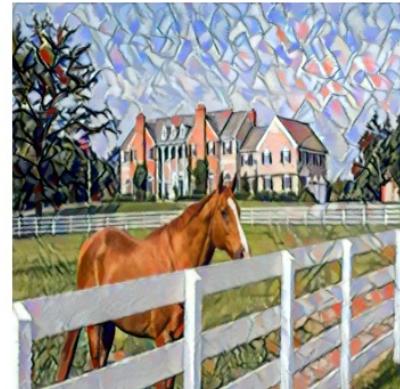
Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0004
Step [100/1000], Content Loss: 0.0071, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0052, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0045, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0040, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0039, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0037, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0035, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0034, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0032, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0005
Step [100/1000], Content Loss: 0.0059, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0042, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0036, Style Loss: 0.0000

Step [400/1000], Content Loss: 0.0032, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0035, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0028, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0032, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0025, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0023, Style Loss: 0.0000

Style Image



Stylized Output

Content Image



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0011
Step [100/1000], Content Loss: 0.0096, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0059, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0048, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0040, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0036, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0034, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0031, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0029, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0028, Style Loss: 0.0000

Style Image



Stylized Output

Content Image



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0157
Step [100/1000], Content Loss: 0.1324, Style Loss: 0.0012
Step [200/1000], Content Loss: 0.0900, Style Loss: 0.0005
Step [300/1000], Content Loss: 0.0702, Style Loss: 0.0003
Step [400/1000], Content Loss: 0.0586, Style Loss: 0.0002
Step [500/1000], Content Loss: 0.0516, Style Loss: 0.0002
Step [600/1000], Content Loss: 0.0470, Style Loss: 0.0001
Step [700/1000], Content Loss: 0.0436, Style Loss: 0.0001
Step [800/1000], Content Loss: 0.0411, Style Loss: 0.0001
Step [900/1000], Content Loss: 0.0389, Style Loss: 0.0001

Style Image



Stylized Output

Content Image



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0026
Step [100/1000], Content Loss: 0.0388, Style Loss: 0.0002
Step [200/1000], Content Loss: 0.0289, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0245, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0221, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0204, Style Loss: 0.0001
Step [600/1000], Content Loss: 0.0190, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0180, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0173, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0167, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0020
Step [100/1000], Content Loss: 0.0192, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0124, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0101, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0088, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0078, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0073, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0068, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0064, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0061, Style Loss: 0.0000

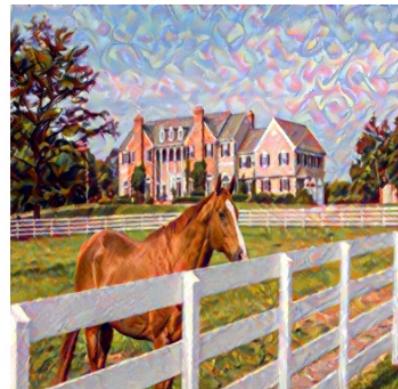
Style Image



Content Image

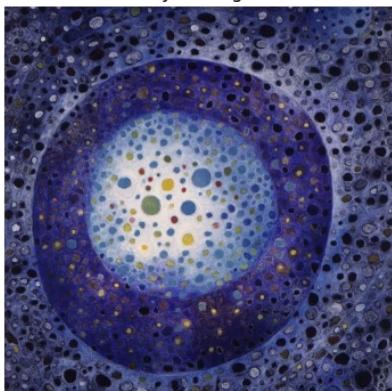


Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0010
Step [100/1000], Content Loss: 0.0148, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0113, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0094, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0083, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0075, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0071, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0068, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0064, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0061, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0006
Step [100/1000], Content Loss: 0.0081, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0061, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0055, Style Loss: 0.0000

Step [400/1000], Content Loss: 0.0051, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0045, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0048, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0041, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0041, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0039, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0006
Step [100/1000], Content Loss: 0.0103, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0077, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0067, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0063, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0062, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0055, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0051, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0051, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0048, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0026
Step [100/1000], Content Loss: 0.0321, Style Loss: 0.0002
Step [200/1000], Content Loss: 0.0236, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0202, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0180, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0165, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0154, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0148, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0140, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0135, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0003
Step [100/1000], Content Loss: 0.0059, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0043, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0041, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0034, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0037, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0030, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0029, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0028, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0032, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0019
Step [100/1000], Content Loss: 0.0387, Style Loss: 0.0004
Step [200/1000], Content Loss: 0.0336, Style Loss: 0.0002
Step [300/1000], Content Loss: 0.0308, Style Loss: 0.0002
Step [400/1000], Content Loss: 0.0285, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0268, Style Loss: 0.0001
Step [600/1000], Content Loss: 0.0254, Style Loss: 0.0001
Step [700/1000], Content Loss: 0.0243, Style Loss: 0.0001
Step [800/1000], Content Loss: 0.0233, Style Loss: 0.0001
Step [900/1000], Content Loss: 0.0226, Style Loss: 0.0001

Style Image



Content Image

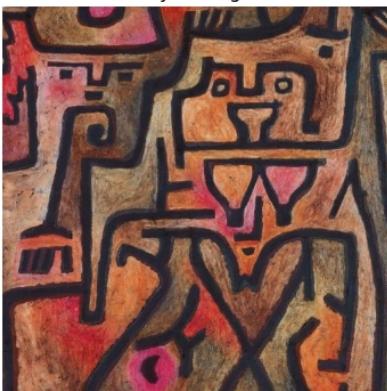


Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0005
Step [100/1000], Content Loss: 0.0073, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0057, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0055, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0047, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0046, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0040, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0039, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0036, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0036, Style Loss: 0.0000

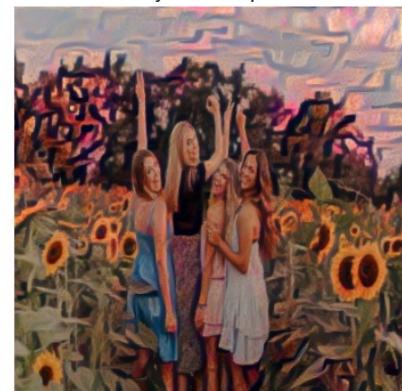
Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0011
Step [100/1000], Content Loss: 0.0154, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0117, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0101, Style Loss: 0.0000

Step [400/1000], Content Loss: 0.0091, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0088, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0084, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0078, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0074, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0073, Style Loss: 0.0000

Style Image



Content Image

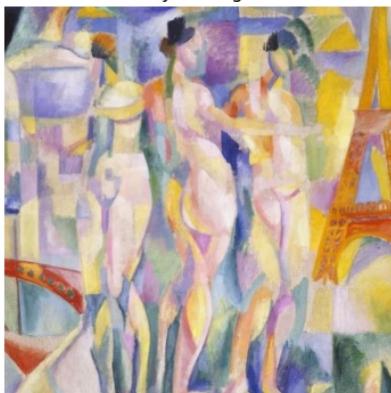


Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0003
Step [100/1000], Content Loss: 0.0046, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0034, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0032, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0025, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0024, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0021, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0022, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0019, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0021, Style Loss: 0.0000

Style Image



Content Image

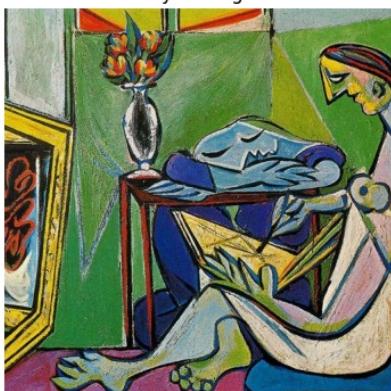


Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0013
Step [100/1000], Content Loss: 0.0130, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0092, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0077, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0067, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0060, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0056, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0055, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0052, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0049, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



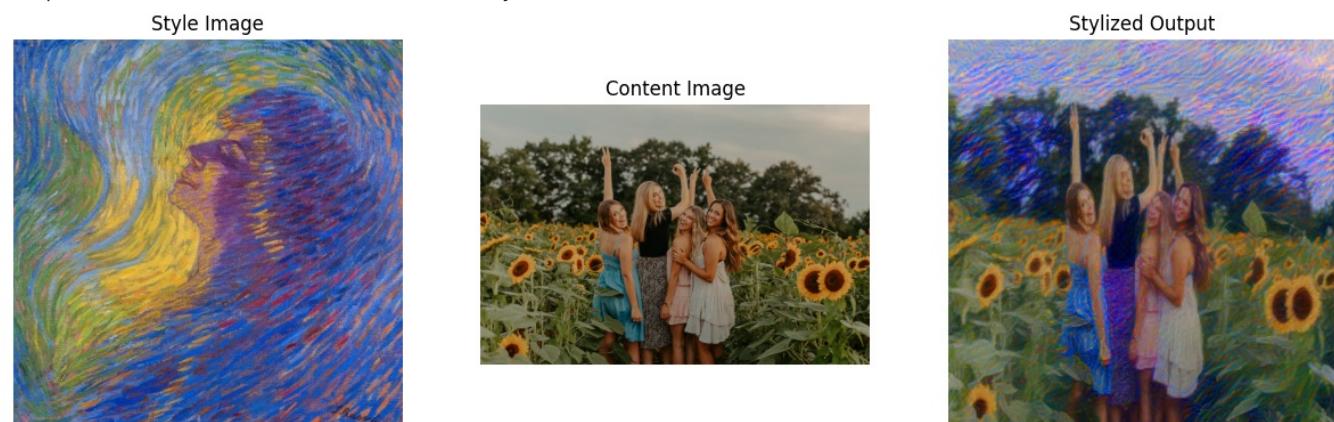
Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0003
Step [100/1000], Content Loss: 0.0063, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0048, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0043, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0044, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0038, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0037, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0041, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0036, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0035, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0030
Step [100/1000], Content Loss: 0.0403, Style Loss: 0.0002
Step [200/1000], Content Loss: 0.0285, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0234, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0205, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0186, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0172, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0162, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0152, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0145, Style Loss: 0.0000

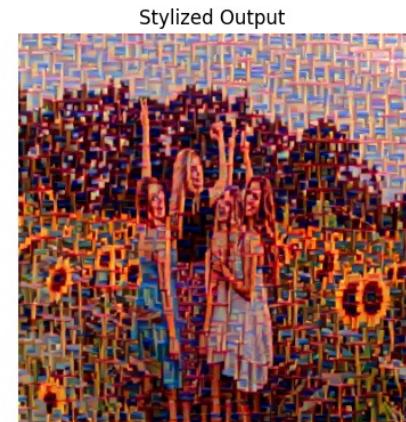
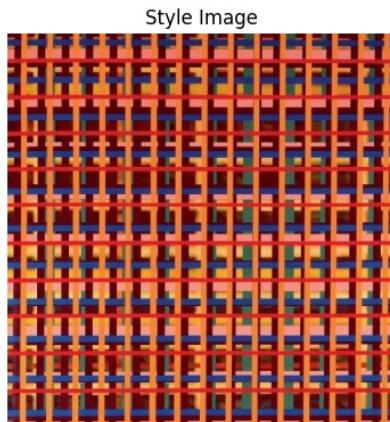


Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0011
Step [100/1000], Content Loss: 0.0110, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0074, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0060, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0054, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0048, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0042, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0039, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0037, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0035, Style Loss: 0.0000

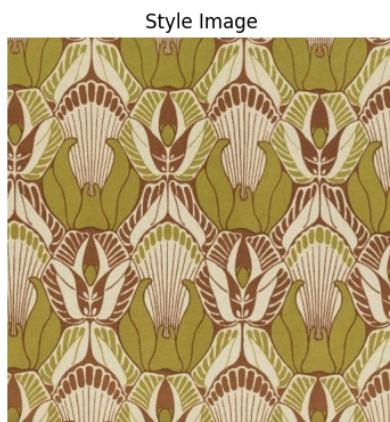


Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0187
Step [100/1000], Content Loss: 0.2649, Style Loss: 0.0019
Step [200/1000], Content Loss: 0.2314, Style Loss: 0.0009
Step [300/1000], Content Loss: 0.2108, Style Loss: 0.0005

Step [400/1000], Content Loss: 0.1975, Style Loss: 0.0004
Step [500/1000], Content Loss: 0.1876, Style Loss: 0.0003
Step [600/1000], Content Loss: 0.1799, Style Loss: 0.0002
Step [700/1000], Content Loss: 0.1731, Style Loss: 0.0002
Step [800/1000], Content Loss: 0.1671, Style Loss: 0.0002
Step [900/1000], Content Loss: 0.1606, Style Loss: 0.0001



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0012
Step [100/1000], Content Loss: 0.0191, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0145, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0127, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0117, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0108, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0104, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0097, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0093, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0090, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0011
Step [100/1000], Content Loss: 0.0084, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0057, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0045, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0039, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0034, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0030, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0029, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0028, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0025, Style Loss: 0.0000



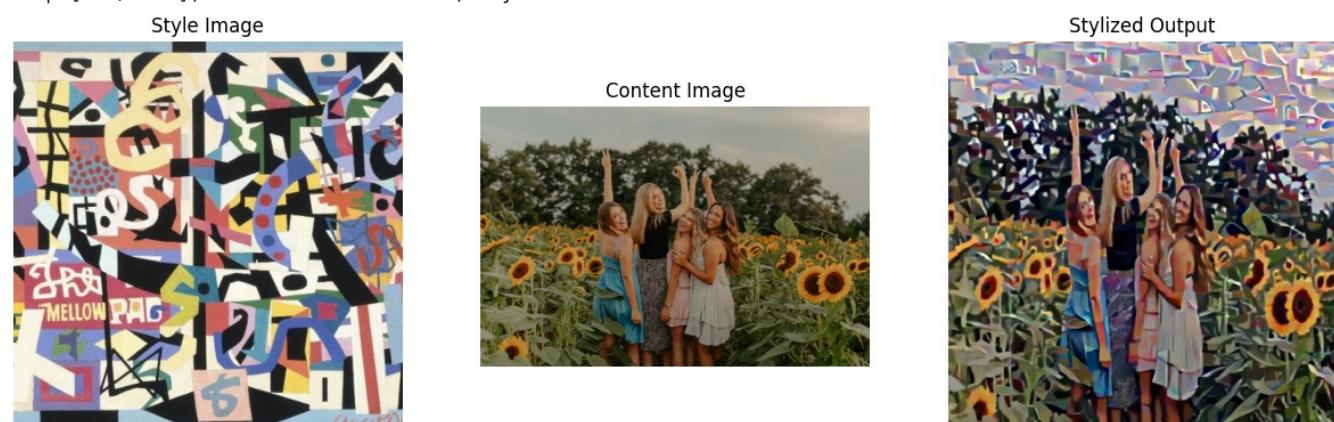
Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0007
Step [100/1000], Content Loss: 0.0105, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0081, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0071, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0067, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0060, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0057, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0054, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0055, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0050, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0021
Step [100/1000], Content Loss: 0.0177, Style Loss: 0.0002
Step [200/1000], Content Loss: 0.0119, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0095, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0086, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0073, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0067, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0063, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0060, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0056, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0025
Step [100/1000], Content Loss: 0.0373, Style Loss: 0.0002
Step [200/1000], Content Loss: 0.0268, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0223, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0195, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0175, Style Loss: 0.0001
Step [600/1000], Content Loss: 0.0162, Style Loss: 0.0001
Step [700/1000], Content Loss: 0.0153, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0145, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0139, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0012
Step [100/1000], Content Loss: 0.0166, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0121, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0103, Style Loss: 0.0000

Step [400/1000], Content Loss: 0.0093, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0085, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0080, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0075, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0071, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0068, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0003
Step [100/1000], Content Loss: 0.0053, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0040, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0036, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0035, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0033, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0032, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0028, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0030, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0028, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0005
Step [100/1000], Content Loss: 0.0051, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0037, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0031, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0029, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0027, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0024, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0022, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0025, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0021, Style Loss: 0.0000



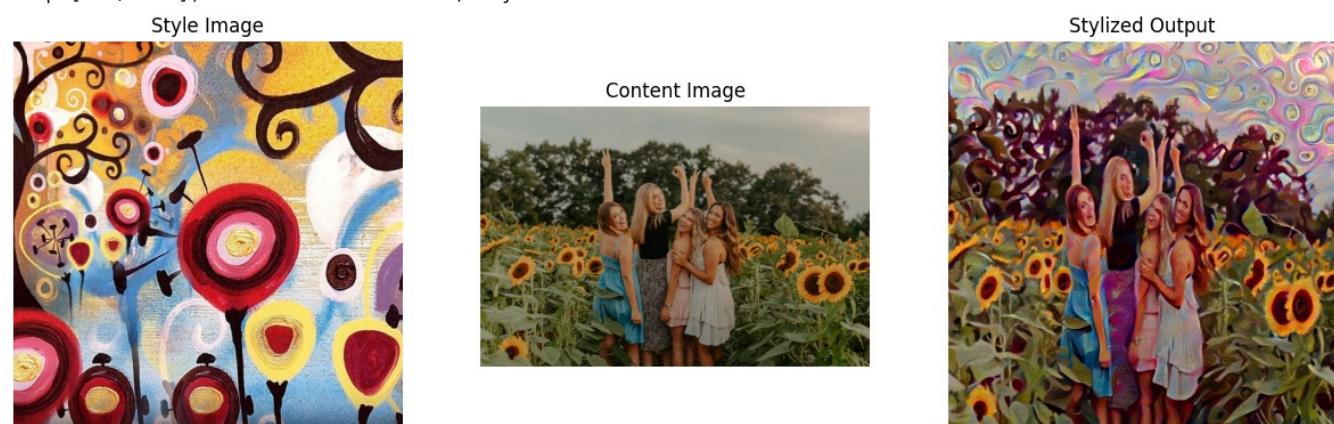
Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0010
Step [100/1000], Content Loss: 0.0055, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0035, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0028, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0023, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0020, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0019, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0017, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0016, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0016, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0151
Step [100/1000], Content Loss: 0.1111, Style Loss: 0.0009
Step [200/1000], Content Loss: 0.0776, Style Loss: 0.0004
Step [300/1000], Content Loss: 0.0615, Style Loss: 0.0002
Step [400/1000], Content Loss: 0.0526, Style Loss: 0.0002
Step [500/1000], Content Loss: 0.0466, Style Loss: 0.0001
Step [600/1000], Content Loss: 0.0421, Style Loss: 0.0001
Step [700/1000], Content Loss: 0.0384, Style Loss: 0.0001
Step [800/1000], Content Loss: 0.0353, Style Loss: 0.0001
Step [900/1000], Content Loss: 0.0326, Style Loss: 0.0001



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0024
Step [100/1000], Content Loss: 0.0329, Style Loss: 0.0002
Step [200/1000], Content Loss: 0.0244, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0210, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0191, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0177, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0167, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0160, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0154, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0150, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0018
Step [100/1000], Content Loss: 0.0142, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0094, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0077, Style Loss: 0.0000

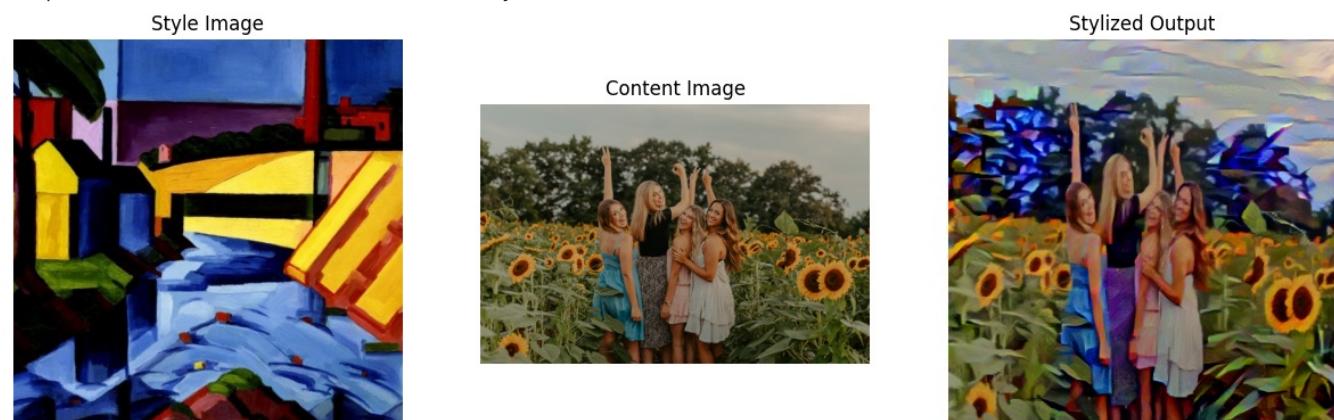
Step [400/1000], Content Loss: 0.0067, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0060, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0057, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0053, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0049, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0051, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0009
Step [100/1000], Content Loss: 0.0105, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0075, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0061, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0053, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0049, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0044, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0041, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0041, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0038, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0006
Step [100/1000], Content Loss: 0.0079, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0061, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0055, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0050, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0046, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0042, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0043, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0041, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0039, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0006
Step [100/1000], Content Loss: 0.0107, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0087, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0078, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0071, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0068, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0066, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0064, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0065, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0061, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0026
Step [100/1000], Content Loss: 0.0274, Style Loss: 0.0002
Step [200/1000], Content Loss: 0.0201, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0166, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0148, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0136, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0128, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0122, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0116, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0111, Style Loss: 0.0000

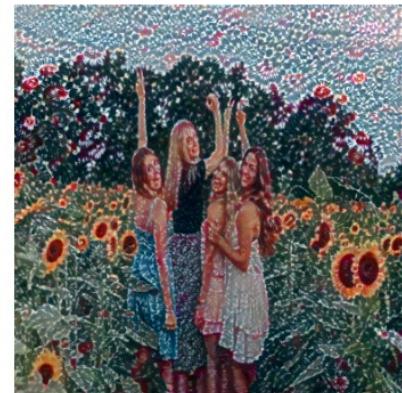
Style Image



Content Image

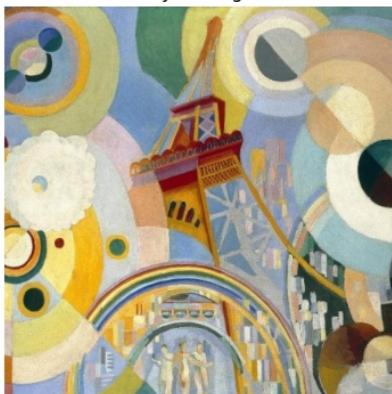


Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0005
Step [100/1000], Content Loss: 0.0067, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0049, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0043, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0040, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0037, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0037, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0034, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0033, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0040, Style Loss: 0.0000

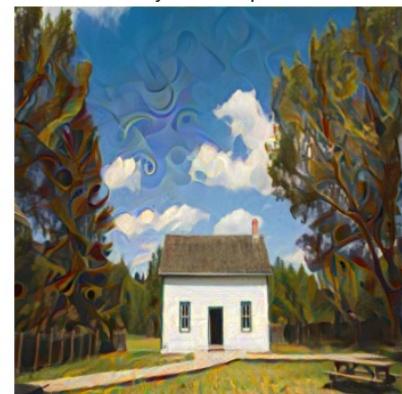
Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0021
Step [100/1000], Content Loss: 0.0338, Style Loss: 0.0004
Step [200/1000], Content Loss: 0.0295, Style Loss: 0.0002
Step [300/1000], Content Loss: 0.0274, Style Loss: 0.0002

Step [400/1000], Content Loss: 0.0260, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0242, Style Loss: 0.0001
Step [600/1000], Content Loss: 0.0231, Style Loss: 0.0001
Step [700/1000], Content Loss: 0.0220, Style Loss: 0.0001
Step [800/1000], Content Loss: 0.0213, Style Loss: 0.0001
Step [900/1000], Content Loss: 0.0207, Style Loss: 0.0001



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0007
Step [100/1000], Content Loss: 0.0096, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0072, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0062, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0055, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0050, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0047, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0046, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0043, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0042, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0013
Step [100/1000], Content Loss: 0.0187, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0143, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0124, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0113, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0106, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0099, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0093, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0089, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0086, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0005
 Step [100/1000], Content Loss: 0.0050, Style Loss: 0.0000
 Step [200/1000], Content Loss: 0.0037, Style Loss: 0.0000
 Step [300/1000], Content Loss: 0.0031, Style Loss: 0.0000
 Step [400/1000], Content Loss: 0.0027, Style Loss: 0.0000
 Step [500/1000], Content Loss: 0.0023, Style Loss: 0.0000
 Step [600/1000], Content Loss: 0.0023, Style Loss: 0.0000
 Step [700/1000], Content Loss: 0.0021, Style Loss: 0.0000
 Step [800/1000], Content Loss: 0.0020, Style Loss: 0.0000
 Step [900/1000], Content Loss: 0.0019, Style Loss: 0.0000

Style Image



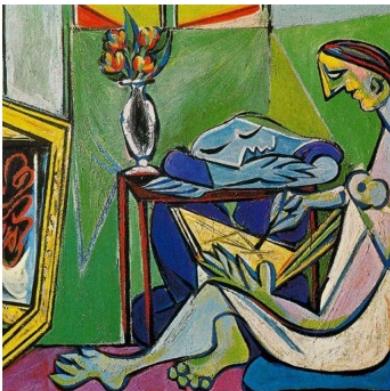
Stylized Output

Content Image



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0015
 Step [100/1000], Content Loss: 0.0146, Style Loss: 0.0001
 Step [200/1000], Content Loss: 0.0111, Style Loss: 0.0000
 Step [300/1000], Content Loss: 0.0095, Style Loss: 0.0000
 Step [400/1000], Content Loss: 0.0085, Style Loss: 0.0000
 Step [500/1000], Content Loss: 0.0079, Style Loss: 0.0000
 Step [600/1000], Content Loss: 0.0074, Style Loss: 0.0000
 Step [700/1000], Content Loss: 0.0071, Style Loss: 0.0000
 Step [800/1000], Content Loss: 0.0068, Style Loss: 0.0000
 Step [900/1000], Content Loss: 0.0066, Style Loss: 0.0000

Style Image



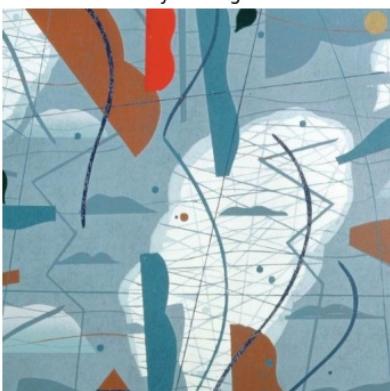
Stylized Output

Content Image



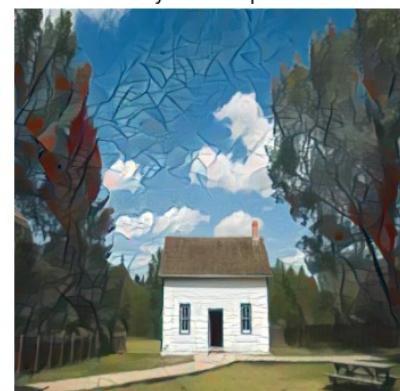
Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0003
 Step [100/1000], Content Loss: 0.0038, Style Loss: 0.0000
 Step [200/1000], Content Loss: 0.0030, Style Loss: 0.0000
 Step [300/1000], Content Loss: 0.0028, Style Loss: 0.0000
 Step [400/1000], Content Loss: 0.0025, Style Loss: 0.0000
 Step [500/1000], Content Loss: 0.0023, Style Loss: 0.0000
 Step [600/1000], Content Loss: 0.0025, Style Loss: 0.0000
 Step [700/1000], Content Loss: 0.0022, Style Loss: 0.0000
 Step [800/1000], Content Loss: 0.0022, Style Loss: 0.0000
 Step [900/1000], Content Loss: 0.0021, Style Loss: 0.0000

Style Image

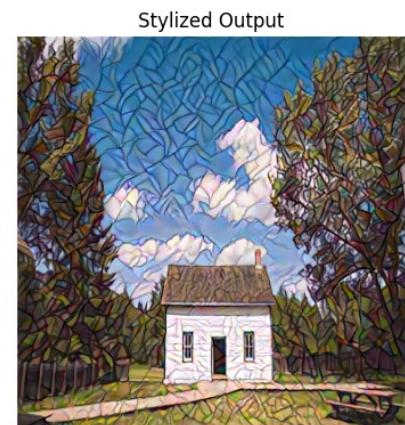
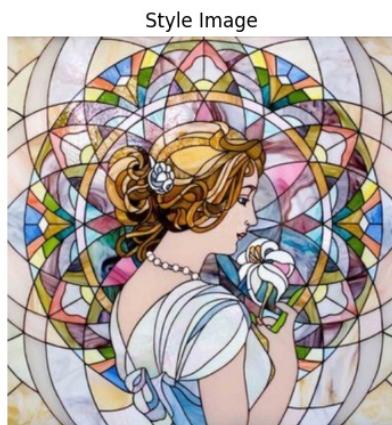


Stylized Output

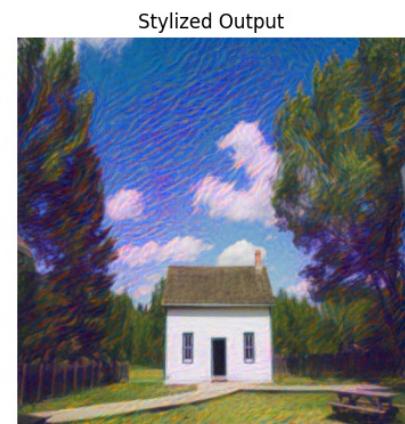
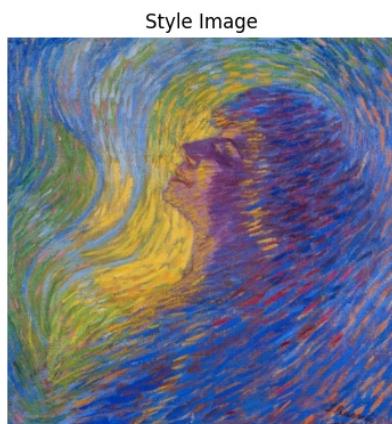
Content Image



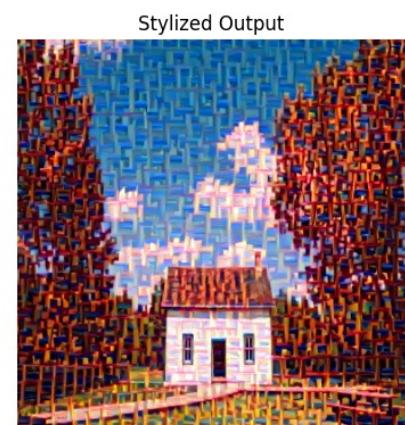
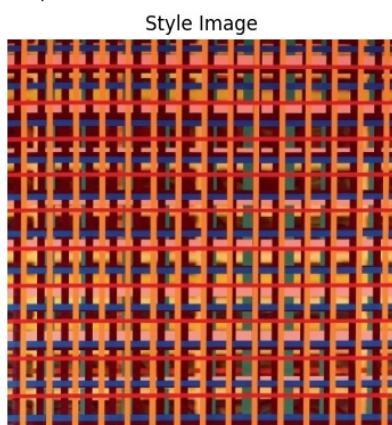
Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0036
 Step [100/1000], Content Loss: 0.0475, Style Loss: 0.0002
 Step [200/1000], Content Loss: 0.0328, Style Loss: 0.0001
 Step [300/1000], Content Loss: 0.0267, Style Loss: 0.0001
 Step [400/1000], Content Loss: 0.0233, Style Loss: 0.0001
 Step [500/1000], Content Loss: 0.0211, Style Loss: 0.0000
 Step [600/1000], Content Loss: 0.0195, Style Loss: 0.0000
 Step [700/1000], Content Loss: 0.0184, Style Loss: 0.0000
 Step [800/1000], Content Loss: 0.0175, Style Loss: 0.0000
 Step [900/1000], Content Loss: 0.0164, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0010
 Step [100/1000], Content Loss: 0.0059, Style Loss: 0.0000
 Step [200/1000], Content Loss: 0.0041, Style Loss: 0.0000
 Step [300/1000], Content Loss: 0.0032, Style Loss: 0.0000
 Step [400/1000], Content Loss: 0.0028, Style Loss: 0.0000
 Step [500/1000], Content Loss: 0.0029, Style Loss: 0.0000
 Step [600/1000], Content Loss: 0.0022, Style Loss: 0.0000
 Step [700/1000], Content Loss: 0.0022, Style Loss: 0.0000
 Step [800/1000], Content Loss: 0.0020, Style Loss: 0.0000
 Step [900/1000], Content Loss: 0.0018, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0198
 Step [100/1000], Content Loss: 0.1762, Style Loss: 0.0025
 Step [200/1000], Content Loss: 0.1472, Style Loss: 0.0011
 Step [300/1000], Content Loss: 0.1308, Style Loss: 0.0007
 Step [400/1000], Content Loss: 0.1199, Style Loss: 0.0005
 Step [500/1000], Content Loss: 0.1117, Style Loss: 0.0004
 Step [600/1000], Content Loss: 0.1061, Style Loss: 0.0003
 Step [700/1000], Content Loss: 0.1013, Style Loss: 0.0002
 Step [800/1000], Content Loss: 0.0974, Style Loss: 0.0002
 Step [900/1000], Content Loss: 0.0940, Style Loss: 0.0002

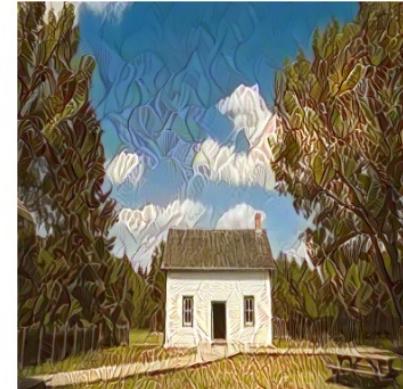


Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0017
Step [100/1000], Content Loss: 0.0240, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0184, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0163, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0149, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0139, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0131, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0125, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0119, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0115, Style Loss: 0.0000

Style Image



Stylized Output

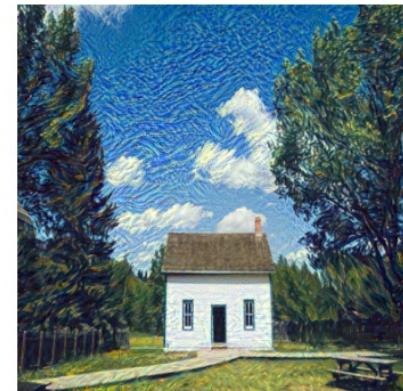
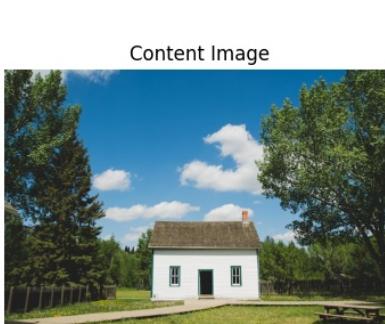


Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0008
Step [100/1000], Content Loss: 0.0042, Style Loss: 0.0000
Step [200/1000], Content Loss: 0.0028, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0023, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0020, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0018, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0017, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0015, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0015, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0015, Style Loss: 0.0000

Style Image



Stylized Output

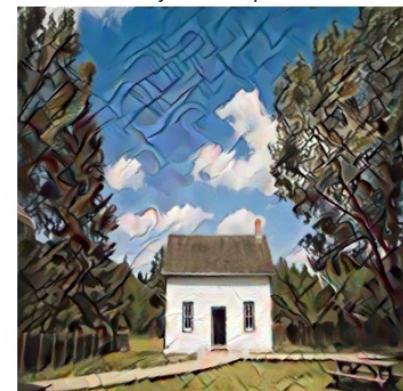
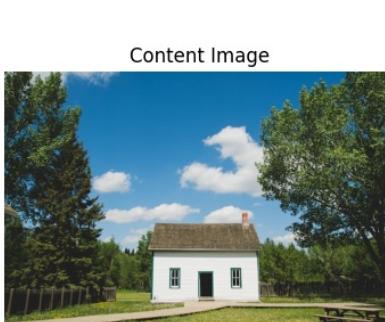


Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0011
Step [100/1000], Content Loss: 0.0131, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0095, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0081, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0067, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0061, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0057, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0054, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0051, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0049, Style Loss: 0.0000

Style Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0028
Step [100/1000], Content Loss: 0.0234, Style Loss: 0.0004
Step [200/1000], Content Loss: 0.0171, Style Loss: 0.0002
Step [300/1000], Content Loss: 0.0140, Style Loss: 0.0001

Step [400/1000], Content Loss: 0.0121, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0109, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0101, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0094, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0090, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0087, Style Loss: 0.0000

Style Image



Stylized Output

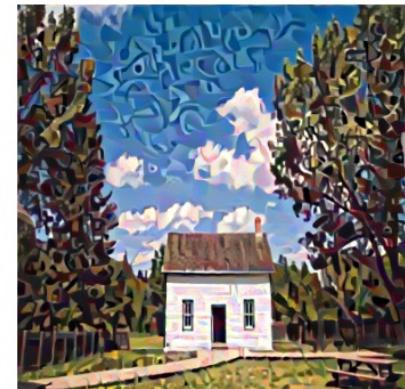


Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0032
Step [100/1000], Content Loss: 0.0364, Style Loss: 0.0003
Step [200/1000], Content Loss: 0.0262, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0221, Style Loss: 0.0001
Step [400/1000], Content Loss: 0.0192, Style Loss: 0.0001
Step [500/1000], Content Loss: 0.0174, Style Loss: 0.0001
Step [600/1000], Content Loss: 0.0162, Style Loss: 0.0001
Step [700/1000], Content Loss: 0.0152, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0143, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0135, Style Loss: 0.0000

Style Image



Stylized Output

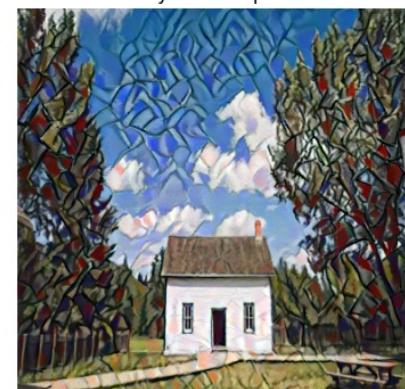


Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0016
Step [100/1000], Content Loss: 0.0206, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0146, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0122, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0108, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0099, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0092, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0089, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0083, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0080, Style Loss: 0.0000

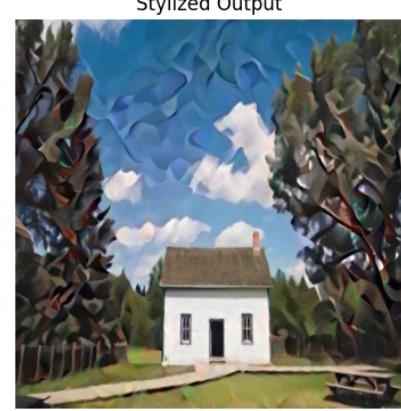
Style Image



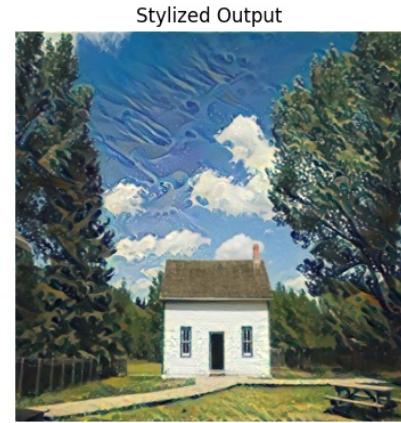
Stylized Output



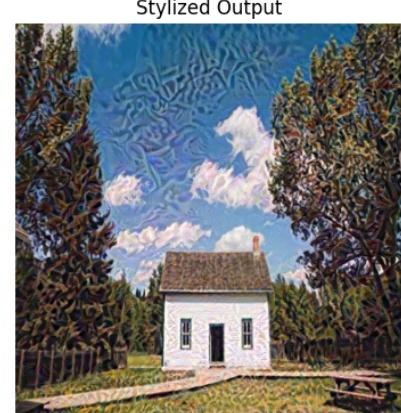
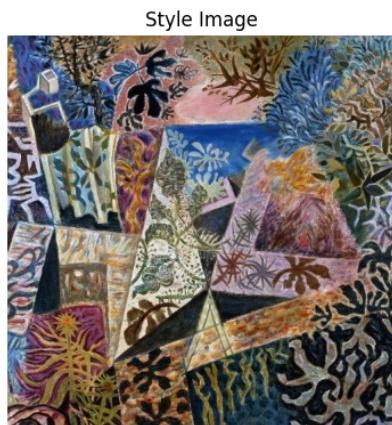
Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0005
 Step [100/1000], Content Loss: 0.0077, Style Loss: 0.0000
 Step [200/1000], Content Loss: 0.0056, Style Loss: 0.0000
 Step [300/1000], Content Loss: 0.0049, Style Loss: 0.0000
 Step [400/1000], Content Loss: 0.0044, Style Loss: 0.0000
 Step [500/1000], Content Loss: 0.0042, Style Loss: 0.0000
 Step [600/1000], Content Loss: 0.0040, Style Loss: 0.0000
 Step [700/1000], Content Loss: 0.0037, Style Loss: 0.0000
 Step [800/1000], Content Loss: 0.0037, Style Loss: 0.0000
 Step [900/1000], Content Loss: 0.0034, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0004
 Step [100/1000], Content Loss: 0.0037, Style Loss: 0.0000
 Step [200/1000], Content Loss: 0.0027, Style Loss: 0.0000
 Step [300/1000], Content Loss: 0.0023, Style Loss: 0.0000
 Step [400/1000], Content Loss: 0.0021, Style Loss: 0.0000
 Step [500/1000], Content Loss: 0.0018, Style Loss: 0.0000
 Step [600/1000], Content Loss: 0.0017, Style Loss: 0.0000
 Step [700/1000], Content Loss: 0.0016, Style Loss: 0.0000
 Step [800/1000], Content Loss: 0.0015, Style Loss: 0.0000
 Step [900/1000], Content Loss: 0.0014, Style Loss: 0.0000

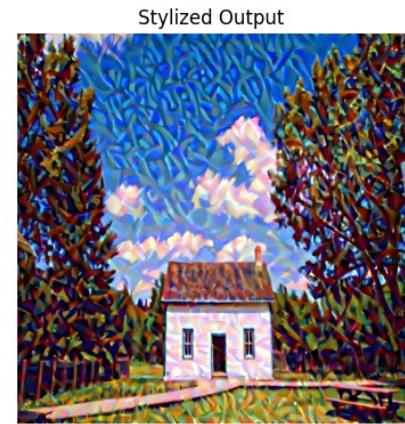
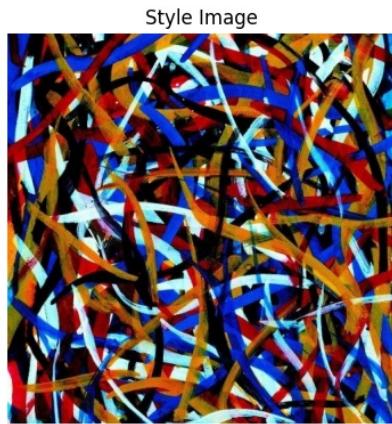


Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0013
 Step [100/1000], Content Loss: 0.0100, Style Loss: 0.0000
 Step [200/1000], Content Loss: 0.0066, Style Loss: 0.0000
 Step [300/1000], Content Loss: 0.0055, Style Loss: 0.0000
 Step [400/1000], Content Loss: 0.0049, Style Loss: 0.0000
 Step [500/1000], Content Loss: 0.0044, Style Loss: 0.0000
 Step [600/1000], Content Loss: 0.0040, Style Loss: 0.0000
 Step [700/1000], Content Loss: 0.0038, Style Loss: 0.0000
 Step [800/1000], Content Loss: 0.0037, Style Loss: 0.0000
 Step [900/1000], Content Loss: 0.0036, Style Loss: 0.0000

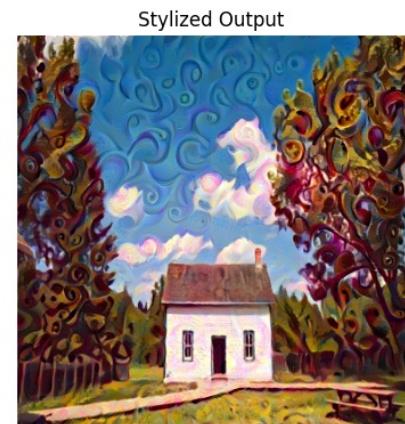
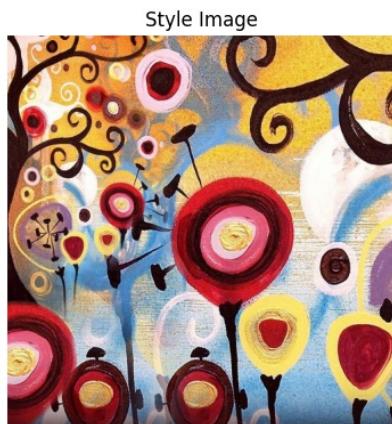


Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0165
 Step [100/1000], Content Loss: 0.1113, Style Loss: 0.0012
 Step [200/1000], Content Loss: 0.0824, Style Loss: 0.0004
 Step [300/1000], Content Loss: 0.0674, Style Loss: 0.0003

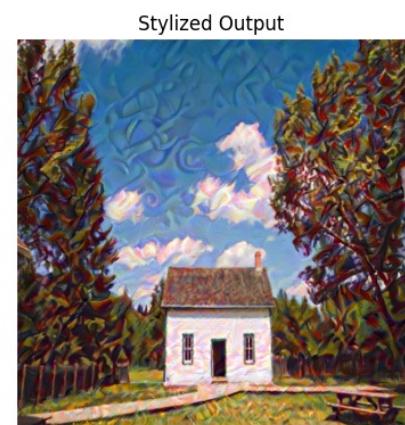
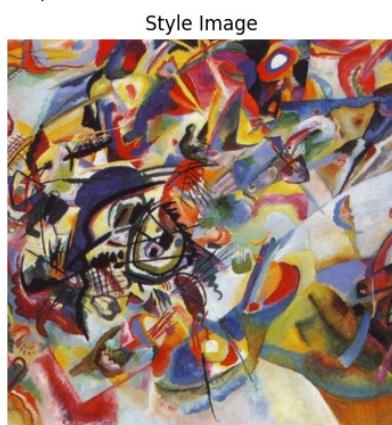
 Step [400/1000], Content Loss: 0.0583, Style Loss: 0.0002
 Step [500/1000], Content Loss: 0.0522, Style Loss: 0.0001
 Step [600/1000], Content Loss: 0.0477, Style Loss: 0.0001
 Step [700/1000], Content Loss: 0.0444, Style Loss: 0.0001
 Step [800/1000], Content Loss: 0.0406, Style Loss: 0.0001
 Step [900/1000], Content Loss: 0.0384, Style Loss: 0.0001



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0031
 Step [100/1000], Content Loss: 0.0382, Style Loss: 0.0003
 Step [200/1000], Content Loss: 0.0278, Style Loss: 0.0001
 Step [300/1000], Content Loss: 0.0241, Style Loss: 0.0001
 Step [400/1000], Content Loss: 0.0218, Style Loss: 0.0001
 Step [500/1000], Content Loss: 0.0202, Style Loss: 0.0001
 Step [600/1000], Content Loss: 0.0191, Style Loss: 0.0000
 Step [700/1000], Content Loss: 0.0182, Style Loss: 0.0000
 Step [800/1000], Content Loss: 0.0174, Style Loss: 0.0000
 Step [900/1000], Content Loss: 0.0169, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0024
 Step [100/1000], Content Loss: 0.0204, Style Loss: 0.0001
 Step [200/1000], Content Loss: 0.0139, Style Loss: 0.0000
 Step [300/1000], Content Loss: 0.0113, Style Loss: 0.0000
 Step [400/1000], Content Loss: 0.0099, Style Loss: 0.0000
 Step [500/1000], Content Loss: 0.0091, Style Loss: 0.0000
 Step [600/1000], Content Loss: 0.0085, Style Loss: 0.0000
 Step [700/1000], Content Loss: 0.0081, Style Loss: 0.0000
 Step [800/1000], Content Loss: 0.0075, Style Loss: 0.0000
 Step [900/1000], Content Loss: 0.0071, Style Loss: 0.0000



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0010
Step [100/1000], Content Loss: 0.0101, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0073, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0060, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0054, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0050, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0047, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0044, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0043, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0041, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0008
Step [100/1000], Content Loss: 0.0092, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0068, Style Loss: 0.0000
Step [300/1000], Content Loss: 0.0059, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0052, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0049, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0046, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0043, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0042, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0039, Style Loss: 0.0000

Style Image



Content Image



Stylized Output



Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0008
Step [100/1000], Content Loss: 0.0113, Style Loss: 0.0001
Step [200/1000], Content Loss: 0.0090, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0079, Style Loss: 0.0000
Step [400/1000], Content Loss: 0.0072, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0068, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0065, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0062, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0060, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0058, Style Loss: 0.0000

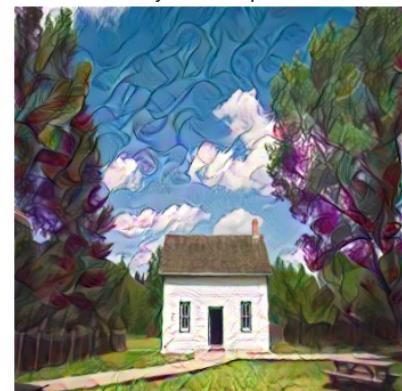
Style Image



Content Image



Stylized Output



```

Step [0/1000], Content Loss: 0.0000, Style Loss: 0.0026
Step [100/1000], Content Loss: 0.0235, Style Loss: 0.0002
Step [200/1000], Content Loss: 0.0173, Style Loss: 0.0001
Step [300/1000], Content Loss: 0.0150, Style Loss: 0.0001

Step [400/1000], Content Loss: 0.0137, Style Loss: 0.0000
Step [500/1000], Content Loss: 0.0127, Style Loss: 0.0000
Step [600/1000], Content Loss: 0.0121, Style Loss: 0.0000
Step [700/1000], Content Loss: 0.0115, Style Loss: 0.0000
Step [800/1000], Content Loss: 0.0111, Style Loss: 0.0000
Step [900/1000], Content Loss: 0.0107, Style Loss: 0.0000

```



Conclusion

In conclusion, the provided Python code demonstrates the implementation of Artistic Style Transfer using the VGG16 model and PyTorch. This technique allows us to combine the content of one image with the artistic style of another, resulting in visually appealing stylized images.

The code leverages the power of pre-trained VGG16, a deep neural network, to extract meaningful features from the content and style images. By freezing the VGG16 parameters during feature extraction, the code focuses solely on preserving the content and transferring the style.

The optimization process involves iteratively updating a generated image using backpropagation. The image is optimized to minimize a combination of content and style losses, where the content loss ensures that the generated image resembles the content image, and the style loss ensures the artistic style transfer from the style image.

The code efficiently handles image preprocessing, Gram matrix calculation, and normalization to produce visually pleasing stylized output images. Each output image is saved with a unique name based on the content and style image combination.

Overall, Artistic Style Transfer is a fascinating application of deep learning and image processing, providing a creative and artistic approach to image manipulation. The code serves as a foundation for exploring and experimenting with different combinations of content and style images to produce unique and captivating results.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js