Lab 4: Neural Style Transfer using VGG19

1. Introduction

Neural style transfer is a technique that combines the content of one image with the style of another using deep neural networks. It leverages pretrained convolutional neural networks (CNNs) to extract content features from the content image and style features from the style image.

In this experiment, we used **VGG19** as the feature extractor. The goal was to stylize **2 content images** using **4 different painting styles each**, and to experiment with the impact of hyperparameters such as style weight, content weight, and number of optimization steps on the stylization output.

Content Images:

- Content_1.jpg
- Content_2.jpg

Style Images:

- Style_1.jpg
- Style_2.jpg
- Style_3.jpg
- Style_4.jpg

Content Images Placeholder
↑ *[IMAGE PLACEHOLDER: Two content images side-by-side]*

Style Images Placeholder
↑ *[IMAGE PLACEHOLDER: Four style images in a 2×2 grid]*

2. Methodology

2.1 Code Structure

- `data_loader.py` : Loads content and style images and converts them to tensors.
- `losses.py` : Implements `ContentLoss` and `StyleLoss` .
- `model.py` : Builds the VGG19 model with inserted loss layers for style and content.
- `utils.py` : Helper functions such as `imshow()` and plotting functions.

- `main.py` : Runs the style transfer process, loops over style images, saves output images, and records loss history.

2.2 Model Architecture

- **Base Network**: Pretrained VGG19
- **Content Layer**: `conv_4`
- **Style Layers**: `conv_1` , `conv_2` , `conv_3` , `conv_4` , `conv_5`

Loss Functions:

- **ContentLoss**: Mean squared error between features of generated image and content image.
- **StyleLoss**: Mean squared error between Gram matrices of generated image and style image.

3. Experiments

3.1 Experiment Setup

- Content Images: 2
- Style Images: 4
- Hyperparameters experimented: `num_steps` , `style_weight` , `content_weight`

3.2 Version 1

Parameters:

- `num_steps` = 400
- `style_weight` = 1e8
- `content_weight` = 5

Observations:

- Stylization effect was minimal; content dominated the output.
- Original images remained mostly unchanged with only faint style influence.
- Training loss ~10, Validation loss ~25 after 50 epochs.
- Both losses stabilized with slight fluctuations.
- Notable fluctuation for the Simon-Lee2 style near epochs 250–300; all other styles stable.

Stylized Outputs (Version 1):

Content Image	Style Image	Output Image

Content Image	Style Image	Output Image
Content_1	Style_1	V1_C1_S1.jpg
Content_1	Style_2	V1_C1_S2.jpg
Content_1	Style_3	V1_C1_S3.jpg
Content_1	Style_4	V1_C1_S4.jpg
Content_2	Style_1	V1_C2_S1.jpg
Content_2	Style_2	V1_C2_S2.jpg
Content_2	Style_3	V1_C2_S3.jpg
Content_2	Style_4	V1_C2_S4.jpg

Version 1 Results Grid
↑[IMAGE PLACEHOLDER: 2×4 grid of Version 1 stylized images]

Version 1 Loss Curves
↑[IMAGE PLACEHOLDER: Loss curves for Content_1 and Content_2 – Version 1]

3.3 Version 2

Parameters:

- `num_steps` = 400
- `style_weight` = 1e8
- `content_weight` = 5

  (Note: Adjust if actual values differ – based on your description, Version 2 used improved settings leading to better results)

Observations:

- Stylization significantly improved compared to Version 1.
- Van Gogh style applied particularly well; others were acceptable.
- Simon-Lee2 style appeared darker and more prominent on the subject.
- Both training and validation losses near 0 and almost overlapping.
- Simon-Lee2 style fluctuated near epoch 400; others stabilized by epoch 30.

Stylized Outputs (Version 2):

Content Image	Style Image	Output Image
Content_1	Style_1	V2_C1_S1.jpg
Content_1	Style_2	V2_C1_S2.jpg

Content Image	Style Image	Output Image
Content_1	Style_3	V2_C1_S3.jpg
Content_1	Style_4	V2_C1_S4.jpg
Content_2	Style_1	V2_C2_S1.jpg
Content_2	Style_2	V2_C2_S2.jpg
Content_2	Style_3	V2_C2_S3.jpg
Content_2	Style_4	V2_C2_S4.jpg

Version 2 Results Grid

↑ [IMAGE PLACEHOLDER: 2×4 grid of Version 2 stylized images – Recommended]

Version 2 Loss Curves

↑ [IMAGE PLACEHOLDER: Loss curves for Content_1 and Content_2 – Version 2]

4. Results

Stylized images clearly demonstrate the effect of style vs. content weights.
Version 2 provides a much better balance between content preservation and style transfer compared to Version 1.
Loss curves confirm convergence and stability of the optimization process.
All output images and plots are saved in:

- Stylized images: `data/output/`
- Loss plots: `Lab_4/MODEL_IMAGES/`

5. Observations & Discussion

- **Version 1**: Content dominates; styles barely visible.
- **Version 2**: Significantly improved stylization; Van Gogh style most prominent, Simon-Lee2 occasionally too dark.
- **Hyperparameter Influence**:
  - Higher `style_weight` → stronger stylization, but can obscure content.
  - Higher `content_weight` → preserves original structure more clearly.
  - More optimization steps → smoother results and better convergence.
  - Deeper layers (e.g., conv_4, conv_5) preserve content structure; earlier layers transfer fine textures.

6. Conclusion

Neural style transfer effectiveness depends heavily on:

- Style/content weight ratio
- Choice of feature layers
- Number of optimization steps

Version 2 achieved a superior visual balance compared to Version 1.

Future Work:

- Experiment with additional content images
- Initialize with random noise for more artistic outputs
- Test different layer combinations for style/content losses
- Implement feed-forward networks (e.g., Perceptual Losses or Fast Style Transfer models) for real-time performance