PROJECT REPORT

NEURAL STYLE TRANSFER

BY: AYUSH KUMAR YADAV
ENROLLMENT NO: 22117039
BRANCH: MECHANICAL

1. Introduction

Style transfer is a long-standing problem that aims to migrate a style from a reference style image to another input picture. This task consists of two steps: (i) texture extraction from the style image and (ii) rendering the content image with the texture. For the first step, lots of methods have been proposed to represent the texture, most of which exploit the deep mid-layer features from the pretrained convolutional neural network (CNN). For the second step, gradient-descent-based methods are used to find an optimal image, which minimizes the distance to both the content image and the style image. Despite the surprising success, these methods need thousands of iterations of gradient descent through large networks for a new input image.

Problem Statement:

Imagine a world where you could transform any ordinary photograph into a masterpiece by applying the artistic style of renowned painters like Van Gogh, Picasso, or Monet. Your task is to develop a neural style transfer model that leverages state-of-the-art deep learning techniques to merge the content of one image with the style of another, creating stunning new visuals that blend the two seamlessly.

This model should be capable of extracting the stylistic features from a given artwork and applying them to a different image while preserving the original content's structure and details. The aim is to achieve a balance between the content and style, resulting in images that are visually appealing and artistically coherent.

Objectives:

1. **Develop a Neural Network for Neural Style Transfer Model:**
   - **Model Architecture:** Design and implement a deep learning model based on CNNs that can perform neural style transfer. This involves selecting an appropriate pre-trained network (such as VGG19) and modifying it to extract relevant content and style features.
   - **Training and Optimization:** Implement optimization algorithms (e.g., gradient descent) to iteratively adjust the generated image so that it minimizes a combined loss function, which incorporates both content and style losses.
   - **Loss Functions**: Define and calculate perceptual loss functions, including content loss (to preserve the structure of the content image) and style loss (to capture the stylistic elements of the style image).
2. **Balance Content and Style:**
   - **Content Preservation:** Ensure that the synthesized image retains the key content and structural elements of the original content image. This involves fine-tuning the content loss weight in the overall loss function.

- **Style Application:** Effectively apply the artistic style from the style image onto the content image. This involves capturing patterns, textures, and colour distributions from the style image using style loss.
- **Hyperparameter Tuning:** Experiment with different weights for content and style losses to achieve the desired balance between content fidelity and stylistic transformation.

3. Create a User-Friendly Interface:

- **User Interface Design:** Develop an intuitive and easy-to-use user interface that allows users to upload their own content and style images.
- **Real-time Processing:** Implement real-time or near-real-time processing capabilities to allow users to see the results of the style transfer quickly.

2. Approach

Steps Involved:

- **Setting Up the Directory:** Set up the path directory to save generated images.

- **Data Preparation**
  - **Collection of Images:**
    - **Content Images:** Gathered a variety of content images representing different subjects and scenes. These images serve as the base upon which the artistic style will be applied.
    - **Style Images:** Collected numerous style images, which include famous artworks, textures, and other visually distinct patterns that will be transferred onto the content images.
  - **Preprocessing:**
    - **Resizing:** All images are resized to a consistent dimension. Using the parameters given below:
      ```
      img_nrows = 400
      img_ncols = int(width*img_nrows/height)
      ```
      This ensures uniformity in the input dimensions for the model.

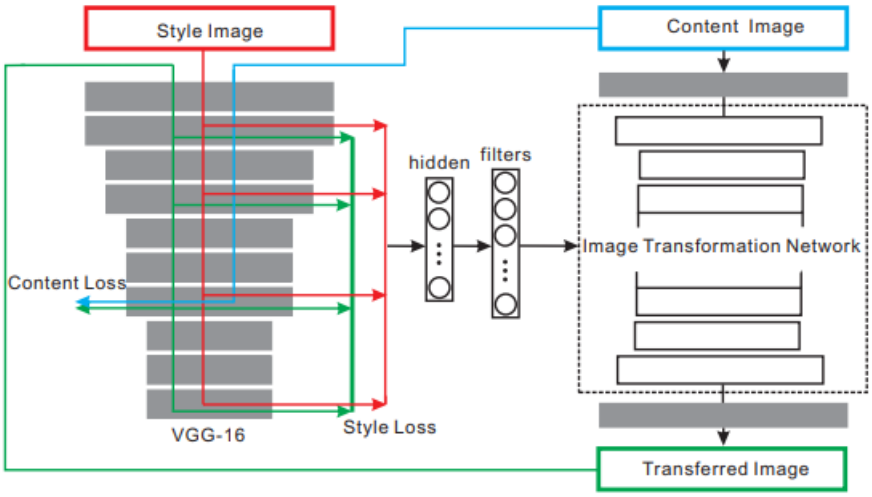
    - **Normalization:** The images are normalized to have pixel values between 0 and 1, which is essential for the proper functioning of neural networks and to match the input requirements of pre-trained models like VGG19.

- **Transfer Learning**
  - Transfer learning with VGG19 involves using its pre-trained convolutional layers to leverage learned features from ImageNet. By removing and replacing the top layers, freezing the base layers, and fine-tuning the new layers on a specific dataset, the model adapts efficiently to new tasks with limited data.
- **Model Architecture**
  - **Base Network - VGG19**:

- **Pre-trained VGG19:** Utilized the VGG19 network, pre-trained on ImageNet, as the backbone for feature extraction. VGG19 is chosen for its proven effectiveness in capturing detailed image features across different layers.
- **Layer Selection:**
  - **Content Layers**: Specifically, 'block5_conv2' is chosen to capture the content information. This layer is deep enough to capture high-level features without losing too much spatial information.
  - **Style Layers:** Selected block1_conv1, block2_conv1, block3_conv1, block4_conv1, and block5_conv1 to capture the style features. These layers span from shallow to deep in the network, allowing for a comprehensive capture of patterns and textures.

- **Loss Functions:**
  - **Content Loss:**
    Defined as the Mean Squared Error (MSE) between the feature representations of the content image and the generated image at the content layer. This ensures that the content of the generated image closely matches the content of the original image.
  - **Style Loss:**
    Calculated using the Gram matrix, which measures the correlations between different filter responses at a given layer. The style loss is the MSE between the Gram matrices of the style image and the generated image across multiple style layers.
  - **Total Variation Loss:**
    Ensures the generated image is smooth.

• Training Process



- **Initialization:**
  The generated image is initialized as a copy of the content image. This initialization helps in preserving the overall structure and content of the content image from the start.

- **Feature Extraction:**
  VGG19 Layers created a custom model to extract outputs from the specified content and style layers. This involves passing the images through the VGG19 network and retrieving the feature maps from the selected layers.

- **Loss Calculation:**

- **Content Loss:** Ensures the generated image retains the structure of the content image. Higher weight on content loss preserves more of the original content.

$$J_{content}(C,G) = \frac{1}{4 \times n_H \times n_W \times n_C} \sum_{\text{all entries}} (a^{(C)} - a^{(G)})^2$$

- **Style Loss:** Ensures the generated image captures the artistic style of the style image. Higher weight on style loss enhances artistic features but may distort content structure.

$$J^{[l]}_{style}(S,G) = \frac{1}{4 \times {n_C}^2 \times (n_H \times n_W)^2} \sum_{i=1}^{n_C} \sum_{j=1}^{n_C} (G^{(S)}_{(gram)i,j} - G^{(G)}_{(gram)i,j})^2$$

- **Gram Matrix:**

$$\mathbf{G}_{gram} = \mathbf{A}_{unrolled} \mathbf{A}^T_{unrolled}$$

- **Optimization:**
  - **Stochastic Gradient Descent (SGD):** Stochastic Gradient Descent (SGD) in Keras is an optimizer that updates model parameters iteratively based on training data. It supports options like learning rate, momentum, and decay for controlling the convergence speed and stability.
  - Using TensorFlow's 'GradientTape', the gradients of the loss with respect to the generated image are computed.
  - The optimizer updates the generated image based on these gradients.

3. Failed Approaches

- **Approach 1: Experimentation with different layers for style extraction**

  Experimentation with different layers for style extraction led to inconsistent outputs.

  Reason for Failure:
  - Inadequate balance between the content and style losses resulted in images that either lost content details or did not adequately reflect the style. Incorrect choice of layers for style feature extraction led to the loss of important stylistic elements.

- **Approach 2: Hyperparameter Tuning**

  Hyperparameters such as the weights of the content and style losses, learning rate, and number of iterations were not properly tuned or were set to default values without optimization.

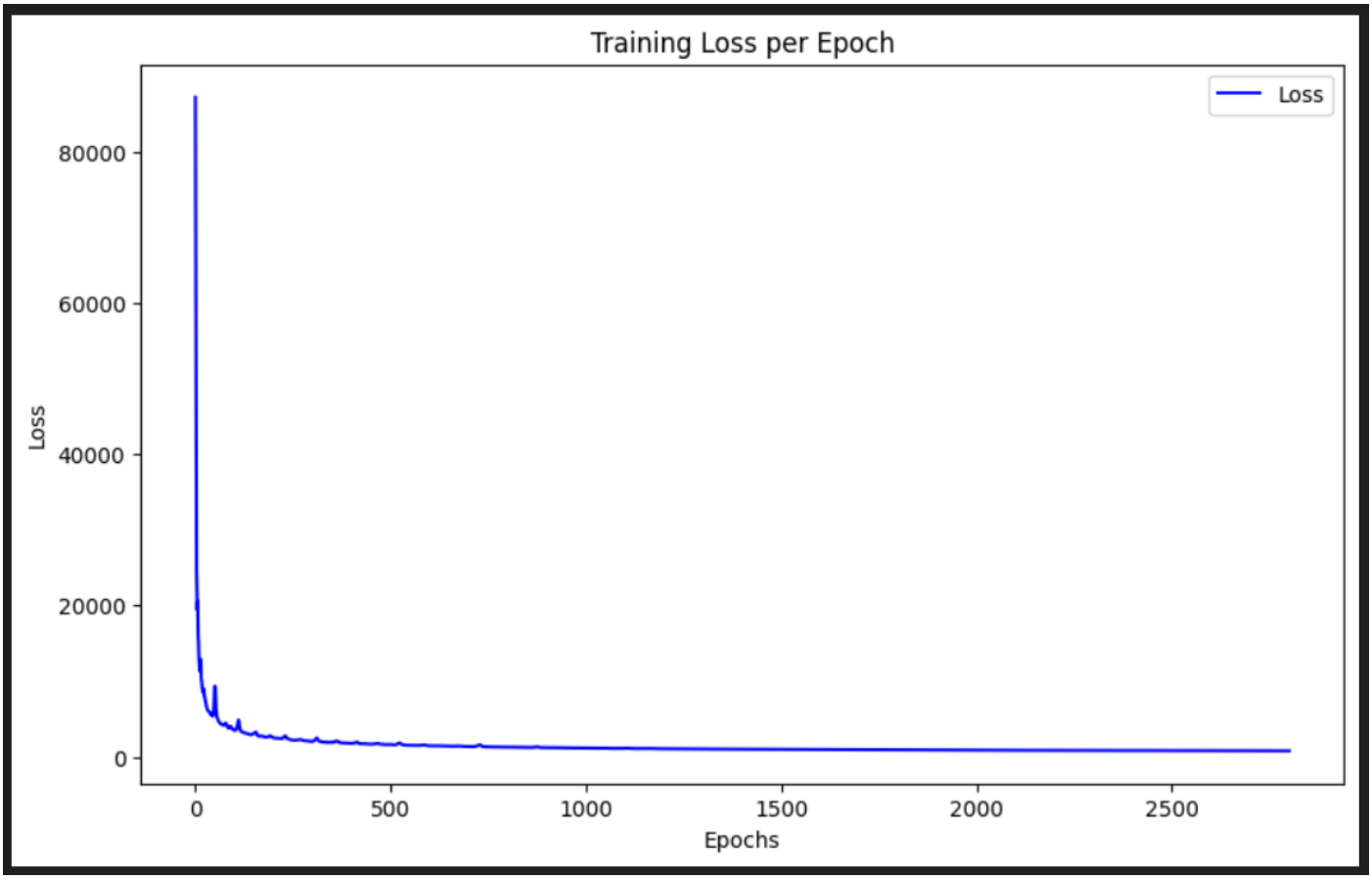
  Reason for Failure:
  - **Imbalanced Stylization:** Incorrect weights for content and style losses lead to generated images that either overly preserve the content at the expense of style or vice versa. The ideal balance between content and style is crucial for coherent results.
  - **Suboptimal Learning Rate:** A learning rate that is too high can cause the generated image to converge too quickly to a suboptimal solution, resulting in artifacts and loss of

details. Conversely, a learning rate that is too low can lead to slow convergence and prolonged training times without significant improvements.

4. Results

Visualizations

Loss Of Features Graph



5. Discussion

- **Significance of Results:**
  - The generated images successfully combine the content of the original images with the artistic style of the style images, demonstrating the effectiveness of neural style transfer
  - Content Preservation: Key features of the content image, such as shapes and forms, were retained, ensuring recognizable content even after stylization.
  - Style Fidelity: Stylistic elements like brush strokes, colour schemes, and textures from the style image were effectively applied to the content image, creating visually striking results.
- **Insights:**
  - The choice of layers for feature extraction significantly impacts the quality of the stylized Image.
  - Proper balancing of content and style losses is crucial to achieve a visually pleasing result.
  - The selection of layers in the VGG19 network for feature extraction significantly impacts the quality of style transfer

6. Conclusion

- **Findings:**
  - This study successfully developed a neural style transfer (NST) model capable of blending the content of one image with the artistic style of another. Leveraging deep learning techniques, specifically the VGG19 convolutional neural network (CNN), the model effectively transferred intricate stylistic elements while preserving the structural integrity of the content images.
  - The generated images consistently demonstrated high aesthetic appeal by seamlessly integrating the stylistic patterns, textures, and colours of the style image with the recognizable shapes and forms of the content image. This achievement underscores the model's ability to produce visually pleasing artworks that mimic the style of renowned artists or artistic genres.
- **Future Improvements:**
  - **Real-Time Style Transfer**: Future research could explore real-time style transfer capabilities using more efficient architectures such as MobileNet. MobileNet's lightweight design could potentially reduce computational overhead, enabling style transfer applications on mobile devices and real-time video processing.
  - **Enhanced User Interface:**
    Incorporating sliders or parameters to adjust the intensity of the style transfer. This feature would allow users to fine-tune the blending effect to better suit their artistic preferences.
    Introducing tools for colour manipulation, such as hue, saturation, and brightness adjustments. This enhancement would provide users with greater flexibility to adapt the stylistic palette to different visual contexts.

7. References

- **Understanding Concept:** https://youtu.be/ZObZRgyZ3Ig?si=EkWTkXlcAlSn-C0X
- **Streamlit Documentation:** https://docs.streamlit.io/
- **Front End Ideation:** https://youtu.be/M3lZNbFJ6I0?si=FgIghyEkrTRIUd-D
- **TensorFlow Documentation:** https://www.tensorflow.org/guide
- **Writing Code:** https://keras.io/examples/generative/neural_style_transfer/#image-preprocessing--deprocessing-utilities
- **Training Image:** https://openaccess.thecvf.com/content_cvpr_2018/papers/Shen_Neural_Style_Transfer_CVPR_2018_paper.pdf