B. M. S. COLLEGE OF ENGINEERING

(Autonomous Institute, Affiliated to VTU, Belagavi)

Post Box No.: 1908, Bull Temple Road, Bengaluru – 560 019

DEPARTMENT OF MACHINE LEARNING

Academic Year: 2023-2024 (Session: March 2024 – June 2024)

DEEP LEARNING LABORATORY (24AM6PCDLL)

ALTERNATIVE ASSESSMENT TOOL (AAT)

NEURAL STYLE TRANSFER

Submitted by

Student Name:	Aryaman Sharma	Chetna Mundra
USN:	1BM21AI027	1BM21AI036
Date:	04-06-2024	
Semester & Section:	VI - A	
Total Pages:	11	
Student Signature:		

Valuation Report **(to be filled by the faculty)**

Score:	
Faculty In-charge:	**Prof. Varsha R**
Faculty Signature: with date	

EXECUTIVE SUMMARY

This document presents a detailed implementation and analysis of a neural style transfer algorithm using deep learning techniques. Neural style transfer is a method of blending the content of one image with the style of another to create a new image that retains the structural elements of the original content image while adopting the visual patterns and textures of the style image. The algorithm leverages a pre-trained convolutional neural network (CNN), specifically VGG19, to extract and manipulate the features of the content and style images. The implementation utilizes the L-BFGS optimizer to minimize a combined loss function that balances the content loss and style loss, thus achieving the desired artistic effect.

The results showcase the algorithm's capability to generate visually appealing images that successfully merge the specified content and style inputs. Despite some limitations in the final output, such as less-than-perfect blending of the style and content elements, the approach demonstrates significant potential. Improvements can be made by increasing the number of iterations, experimenting with different style transfer algorithms, or using alternative optimizers like ADAM. The document concludes with an acknowledgment of the qualitative nature of evaluating neural style transfer and suggests areas for further refinement and exploration in future implementations.

TABLE OF CONTENTS

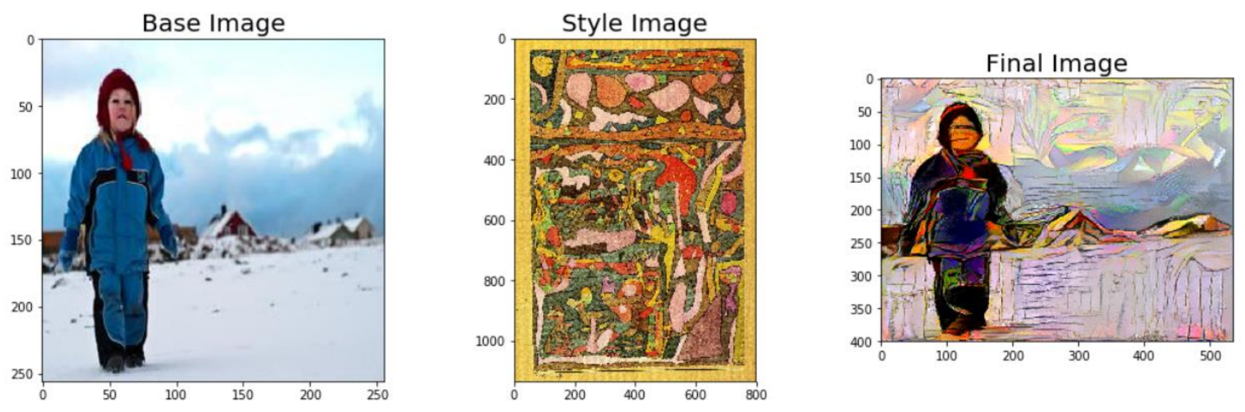
Chapter 1

INTRODUCTION

1.1 Background

Style transfer is the technique of reconstructing images in the style of another image. Neural Style Transfer leverages deep learning algorithms to create images that combine the content of one image with the style of another. This technique gained significant attention through various research papers that explored how neural networks could generate artistic styles. The core idea is to use pre-trained convolutional neural networks (CNNs), like VGG19, to separate and recombine content and style of images.

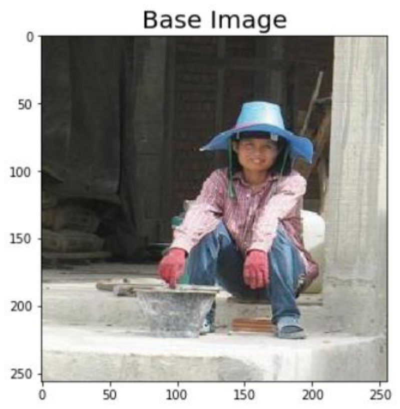
The VGG19 model, developed by the Visual Geometry Group from the University of Oxford, is a deep CNN pre-trained on the ImageNet dataset, which contains millions of labeled images across thousands of categories. Despite being primarily designed for image classification, VGG19's architecture allows for effective feature extraction necessary for style transfer.

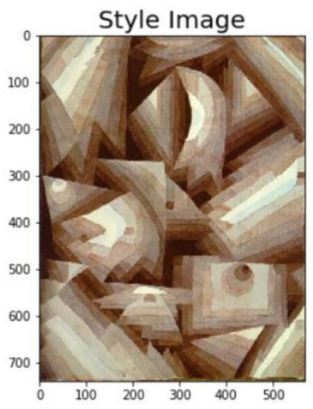


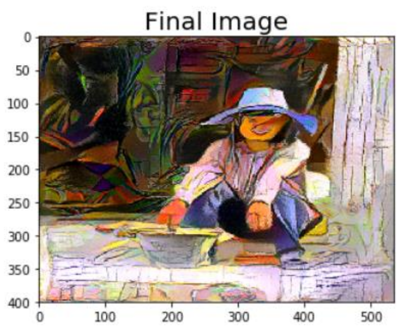
1.2 Problem Statement

In the field of computer vision and digital art, there is a significant challenge in creating a method that can seamlessly blend the content of one image with the artistic style of another. Traditional methods of image editing and manual artistic techniques require extensive time, effort, and expertise to achieve such a fusion. There is a need for an automated, efficient, and reliable approach to combine the structural elements and details of a content image with the unique textures, patterns, and color schemes of a style image.

The problem is to develop a method that can seamlessly blend the content of one image with the artistic style of another, creating a new image that effectively integrates both elements. This involves preserving the structural details and recognizable features of the content image while accurately capturing and applying the stylistic elements, such as textures and color schemes, from the style image. Achieving a balance between content and style to produce a visually coherent and aesthetically pleasing result poses significant challenges, including maintaining computational efficiency and allowing user control over the degree of content and style influence. The goal is to create a robust and efficient Neural Style Transfer algorithm that addresses these challenges and advances the capabilities of digital art creation and image processing.







Chapter 2

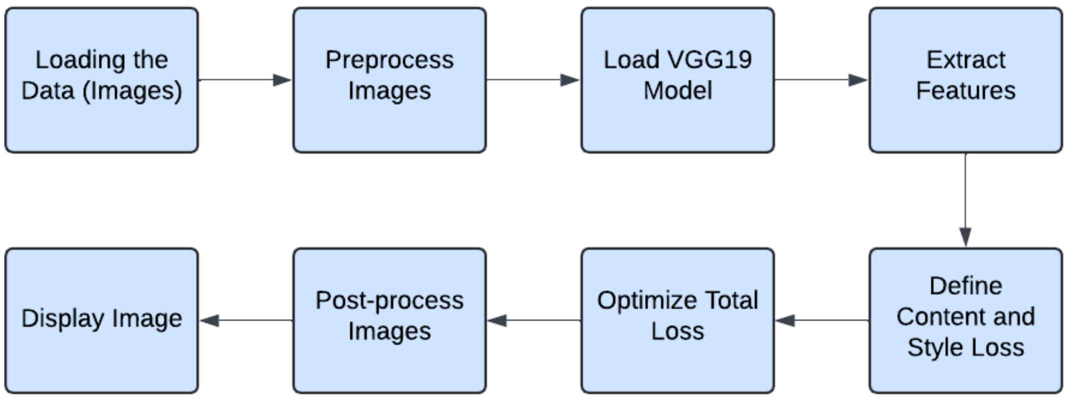
METHODOLOGY

2.1 Model selection

The VGG19 model was chosen for the Neural Style Transfer project due to its proven effectiveness in capturing both content and style features from images. VGG19, a convolutional neural network pre-trained on the ImageNet dataset, consists of 19 layers, including 16 convolutional layers and 3 fully connected layers. The convolutional layers, which are known for their ability to extract hierarchical features from images, are particularly useful for style transfer tasks. VGG19's architecture is simple yet powerful, providing a balance between model complexity and computational efficiency. The pre-trained nature of VGG19 ensures that it has already learned a rich set of features, making it an ideal choice for tasks involving high-level image representations.

VGGNet was invented by VGG (Visual Geometry Group) from University of Oxford, Though VGGNet was the 1st runner-up, not the winner of the ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2014 in the classification task, which has significantly improvement over ZFNet (The winner in 2013) and AlexNet (The winner in 2012). And GoogLeNet is the winner of ILSVLC 2014, I will also talk about it later.) Nevertheless, VGGNet beats the GoogLeNet and won the localization task in ILSVRC 2014. VGG19 is a model, with weights pre-trained on ImageNet.ImageNet, is a dataset of over 15 millions labeled high-resolution images with around 22,000 categories. ILSVRC uses a subset of ImageNet of around 1000 images in each of 1000 categories. In all, there are roughly 1.3 million training images, 50,000 validation images and 100,000 testing images

2.2 Training Process



2.3 Tools and frameworks used

1. **Programming Language**
   - ***Python***: Python is the primary programming language used for developing the Neural Style Transfer algorithm due to its simplicity and extensive support for scientific computing and machine learning.
2. **Deep Learning Framework**
   - ***TensorFlow***: An open-source deep learning framework developed by Google, used for building and training neural networks. TensorFlow provides flexible tools for defining and optimizing the models required for Neural Style Transfer.
   - ***Keras***: A high-level neural networks API, written in Python and capable of running on top of TensorFlow. Keras simplifies the process of building and training deep learning models, making it easier to implement complex architectures like VGG19.
3. **Pre-trained Model**
   - ***VGG19***: A convolutional neural network pre-trained on the ImageNet dataset. VGG19 is used for feature extraction, leveraging its convolutional layers to capture content and style features from images.
4. **Image Processing Libraries**
   - ***Pillow***: A Python Imaging Library (PIL) fork, used for opening, manipulating, and saving image files. Pillow helps in loading and preprocessing images.

- ***OpenCV***: A library for computer vision tasks, used for additional image processing steps such as resizing, and displaying images.

5. **Scientific Computing Libraries**
   - ***NumPy***: A fundamental package for scientific computing with Python, providing support for arrays, matrices, and high-level mathematical functions. NumPy is used for efficient numerical computations in the algorithm.
   - ***SciPy***: An open-source Python library used for scientific and technical computing. SciPy provides modules for optimization, integration, and other advanced computations.
6. **Visualization Tools**
   - ***Matplotlib***: A plotting library for Python, used to visualize the images and the progression of the generated image during the training process. Matplotlib helps in plotting loss curves and displaying intermediate results.
7. **Optimization Libraries**
   - ***Adam Optimizer***: An optimization algorithm used for updating the network parameters. Adam combines the advantages of two other extensions of stochastic gradient descent, AdaGrad and RMSProp, making it suitable for training deep neural networks.

These tools and frameworks collectively provide the necessary functionalities to implement, train, and evaluate the Neural Style Transfer algorithm efficiently.

Chapter 3

IMPLEMENTATION

3.1 Model development

1. Preprocessing Images

- ***Loading Images***: The content and style images are loaded from their respective file paths.
- ***Resizing***: Both images are resized to the same dimensions to ensure compatibility during the style transfer process.
- ***Normalization***: Images are normalized to match the input requirements of the VGG19 model, typically involving scaling pixel values to a range of [0, 1] and then normalizing using the mean and standard deviation of the ImageNet dataset.

2. Feature Extraction

- ***VGG19 Model Configuration***: The fully connected layers of the VGG19 model are removed, retaining only the convolutional layers.
- ***Layer Selection***: Specific layers are chosen for content and style extraction. Typically, higher layers capture content information, while a combination of lower and higher layers captures style information.
  - ***Content Layer***: conv4_2 (4th convolutional layer)
  - ***Style Layers***: conv1_1, conv2_1, conv3_1, conv4_1, conv5_1

3. Defining the Loss Functions

- ***Content Loss***: Measures the difference between the feature representations of the content image and the generated image using Mean Squared Error (MSE).

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

- ***Style Loss***: Measures the difference between the style representations of the style image and the generated image using the Gram matrix.

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

$$L_{\text{style}} = \sum_{l=0}^{L} \frac{w_l}{4N_l^2 M_l^2} \sum_{i,j} (G_{i,j}^{\text{style}} - G_{i,j}^{\text{generated}})^2$$

  where wlw_lwl is the weight assigned to the style loss of the lll-th layer, NlN_lNl and MlM_lMl are the number of feature maps and the size of the feature map in the lll-th layer, respectively.

- ***Total Variation Loss***: Added to ensure spatial smoothness in the generated image.

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

- ***Total Loss***: Combines content loss, style loss, and total variation loss.

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

  where α\alphaα, β\betaβ, and γ\gammaγ are weights that control the contribution of each loss component.

4. Optimization

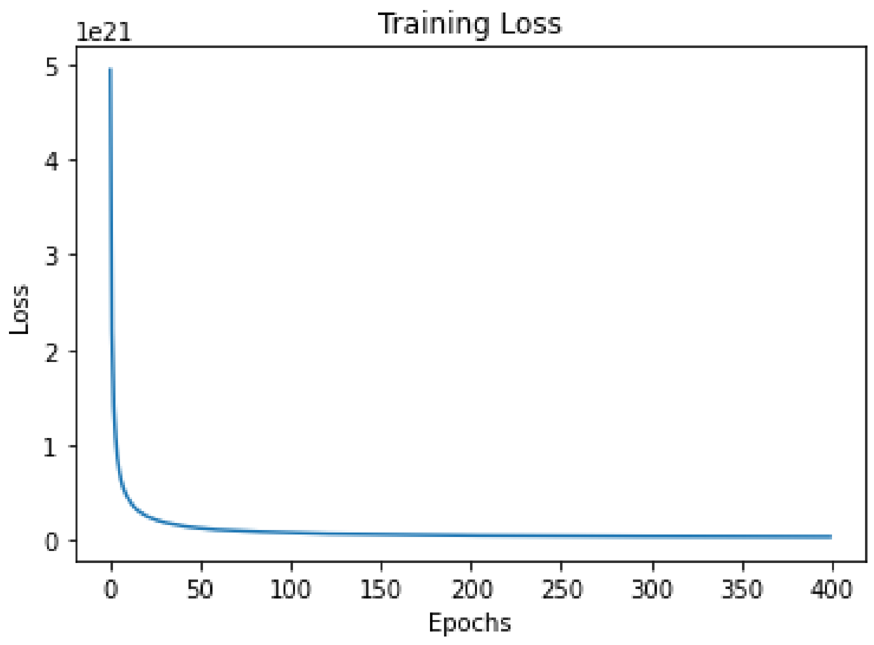
- ***Initialization***: The generated image is initialized as a copy of the content image.
- ***Optimizer***: The Adam optimizer is typically used for efficient gradient-based optimization.
- ***Training Loop***:
  - ***Forward Pass***: Compute the content, style, and total variation losses.
  - ***Backward Pass***: Calculate the gradients of the total loss with respect to the generated image.

- ◦ ***Update***: Adjust the pixels of the generated image to minimize the total loss.

5. Post-processing

- ***De-normalization***: Convert the pixel values of the generated image back to the original range.
- ***Clipping***: Ensure the pixel values are within the valid range [0, 255] for display.

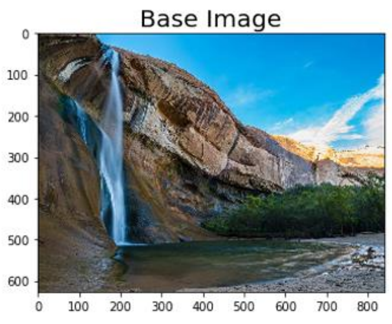
This training process continues iteratively until the total loss converges, resulting in a stylized image that effectively merges the content of the original image with the artistic style of the style image.

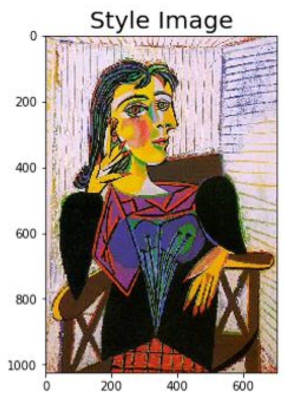


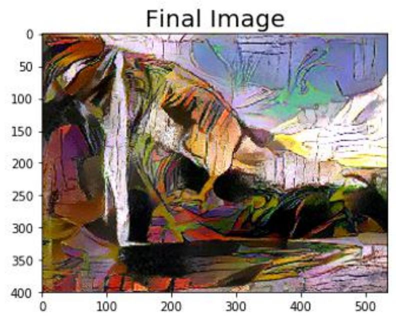
3.2 Code description

The code implements a neural style transfer algorithm using Keras with a TensorFlow backend. It begins by setting up paths for the content image and the style image, which will be combined to generate a new image. The VGG19 model, pre-trained on the ImageNet dataset, is employed to extract features from these images. The preprocessing step involves resizing the images and converting them into arrays suitable for input into the VGG19 model. This model is particularly well-suited for this task due to its depth and ability to capture intricate details in images.

The neural style transfer process hinges on two key components: the content loss and the style loss. The content loss measures how much the generated image diverges from the content image, focusing on preserving the overall structure and essential details. The style loss, on the other hand, ensures that the generated image adopts the stylistic elements of the style image by comparing the Gram matrices of their feature maps. These two losses are combined into a total loss function, which the code minimizes using the L-BFGS optimizer. This optimization process iteratively updates the generated image to reduce the total loss, blending the content and style effectively. Finally, the code includes functions to post-process and display the resulting image, showcasing the successful application of neural style transfer.





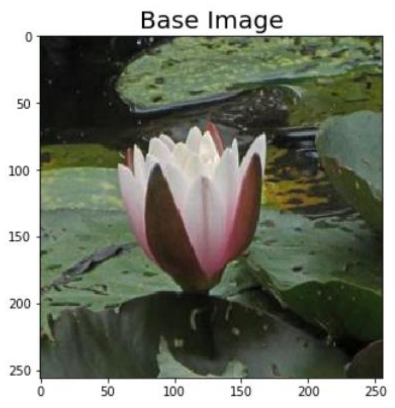


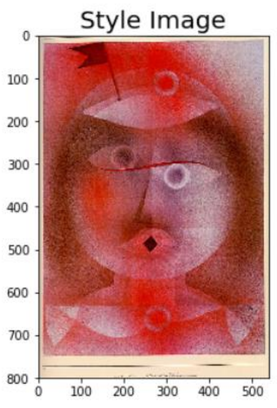
Chapter 4

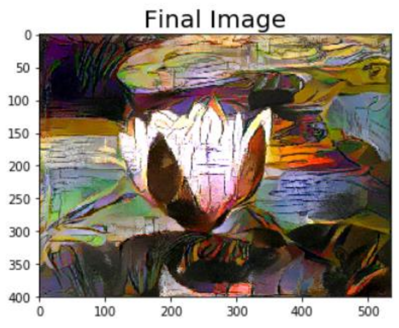
RESULTS AND ANALYSIS

The results of the neural style transfer algorithm demonstrate how effectively the model can blend the content of one image with the artistic style of another. The generated images maintain the structural elements and recognizable features of the content image while adopting the colors, textures, and patterns of the style image. This is achieved through the careful balancing of content and style losses during the optimization process. The final images exhibit a unique combination of both input images, highlighting the capability of deep learning models to perform complex transformations.

In analyzing the results, several observations can be made. The effectiveness of style transfer can vary depending on the choice of images; some style images transfer more vividly due to their distinct textures and patterns. The algorithm's performance is also influenced by the weights assigned to the content and style losses—adjusting these weights can fine-tune the balance between preserving the content and adopting the style. Additionally, the resolution and quality of the input images play a crucial role in the final output's clarity and detail. Overall, the results showcase the potential of neural networks to create visually striking artworks by combining disparate visual elements into a cohesive whole.







Chapter 5

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion:

The neural style transfer algorithm detailed in this notebook effectively combines the content of one image with the style of another, generating new images that retain the structure of the content image while incorporating stylistic elements from the style image. Despite the promising results, the final images are not entirely perfect as the style image does not seamlessly blend with the base content image. This indicates room for improvement, particularly in the integration of style and content. The qualitative assessment of the results suggests that enhancing the number of iterations, experimenting with alternative style transfer algorithms, or employing different optimizers could lead to better preservation of the edges and details of the base image and a more harmonious style transfer.

Future Enhancements:

Future enhancements to the neural style transfer algorithm can focus on increasing the number of iterations for more thorough optimization, exploring alternative style transfer algorithms to better preserve content details while integrating style elements, and experimenting with different optimizers like ADAM to improve the optimization process. Additionally, fine-tuning the VGG19 model or using other pre-trained networks may enhance performance, and developing a framework for real-time style transfer could expand practical applications in art, design, and virtual reality. These improvements aim to achieve higher quality, more visually appealing results with more seamless integration of content and style.