**Neural Style Transfer Experiments with TensorFlow and VGG19**

**Author:** [Keyur Vaidya](mailto:keyur23@bu.edu)

## 

## **Introduction**

In the realm of computer vision and artificial intelligence, style transfer stands out as a fascinating technique that merges the distinct visual characteristics of one image (the style) with the content of another. This fusion creates unique, artistically transformed images, opening avenues for creativity and experimentation in digital art. At the heart of our experiment lies TensorFlow, an open-source platform known for its flexibility and robustness in machine learning and deep learning applications. Leveraging TensorFlow, we utilize the VGG19 model, a convolutional neural network pre-trained on the ImageNet dataset. VGG19 is renowned for its effectiveness in image recognition and feature extraction, making it an ideal choice for style transfer tasks. Our objective in this project is to harness the capabilities of TensorFlow and VGG19 to perform style transfer, aiming to seamlessly blend the style features from one image with the content attributes of another, thereby creating visually compelling and stylistically enriched composite images. This experiment not only showcases the power of neural networks in artistic endeavors but also explores the boundaries of machine learning in creative processes.

## **Methodology**

#### Data Preparation

* Image Selection: Two sets of images were chosen for the trials:
  + First Trial: harry style.jpeg (Content Image) and princetest1.jpeg (Style Image).
  + Second Trial: TRY 2 SELENA.jpeg (Content Image) and Test2.jpeg (Style Image).
* Image Processing: Images were preprocessed to fit the input requirements of VGG19. This involved resizing images to a common dimension and converting them into tensor format suitable for neural network processing.

#### Model Configuration

* VGG19 Model: A convolutional neural network pre-trained on ImageNet, known for its high accuracy in image recognition tasks.
* Relevance for Style Transfer: VGG19 effectively captures image features at various abstraction levels, making it suitable for separating and recombining content and style of different images.

#### Parameter Setting

* Weights Assignment:
  + Style Weight: 1*e*−6
  + Content Weight: 2.5*e*−8
  + Total Variation Weight: 1*e*−6
* Rationale: These weights balance the emphasis on style, content, and smoothness in the generated image.
* Image Dimensions: Determined dynamically based on the aspect ratio of the content image, with a fixed height of 400 pixels to standardize input size while maintaining aspect ratio.

| Table 1: Weight Parameters and Their Descriptions | |
| --- | --- |
| Parameter | Description |
| Style Weight | Controls the degree of styling applied from the style image. |
| Content Weight | Preserves the content of the base image in the generated image. |
| Total Variation Weight | Ensures smoothness and coherence in the generated image. |

*Table 1: Weight Parameters and Their Descriptions*

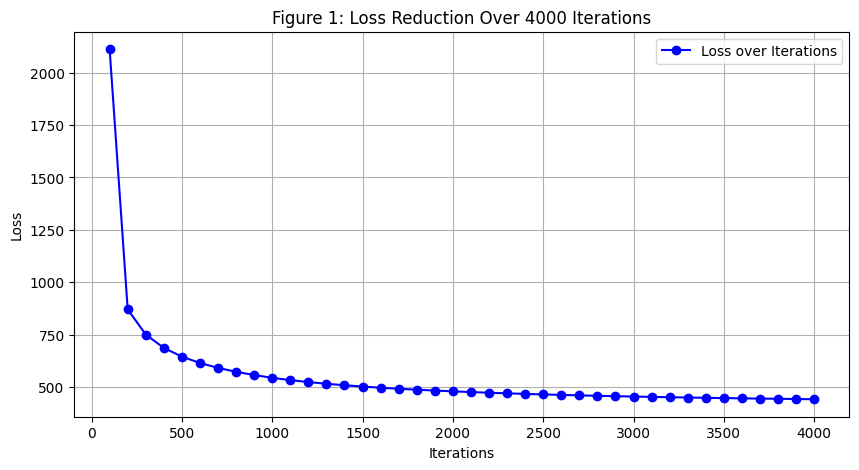
### **Implementation**

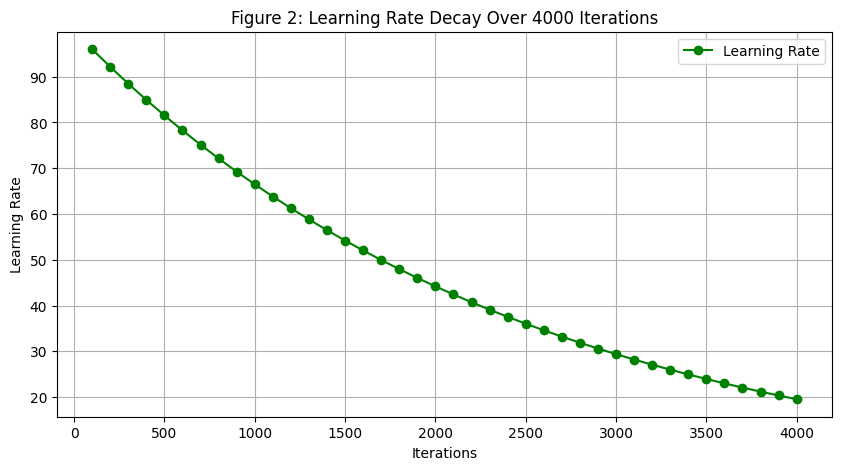
#### *Code Overview*

* *preprocess\_image: This function prepares images for processing by resizing them to a uniform dimension and converting them into tensor format suitable for input into the VGG19 model. It also applies necessary preprocessing steps for VGG19.*
* *deprocess\_image: Converts tensors back into images, reversing the transformations done during preprocessing, which includes reordering color channels and adding mean pixel values.*
* *gram\_matrix: A crucial component in style transfer, this function calculates the Gram matrix of an image tensor, capturing the style by emphasizing feature correlations within layers of the neural network.*
* *style\_loss: Computes the loss between the style of the reference image and the generated image, emphasizing the preservation of style features.*
* *content\_loss: Measures the difference in content between the base and generated images, ensuring the original content's integrity.*
* *total\_variation\_loss: Aims to encourage spatial smoothness in the generated image, reducing noise and enhancing visual quality.*
* *Model Setup: Utilizes VGG19, a pre-trained model, as a feature extractor to access intermediate layers necessary for style and content representation.*

#### *Iteration Process*

* *Optimizer: Utilizes the Stochastic Gradient Descent (SGD) optimizer with an Exponential Decay learning rate schedule. This choice helps in gradually converging to a solution by adjusting the learning rate over iterations.*
* *Iterations (4000): A higher number of iterations allows for gradual and more nuanced refinement of the generated image. The process iteratively updates the image to minimize the loss functions.*
* *Learning Rate & Decay: The initial high learning rate (100.0) expedites early progress, while decay ensures finer adjustments in later stages. The decay rate of 0.96 every 100 steps balances swift initial changes with careful fine-tuning in subsequent iterations.*

*Figure 1: Loss Reduction Over Iterations  
[Insert graph of loss reduction over 4000 iterations]*

*Figure 2: Learning Rate Decay  
[Insert graph showing exponential decay of learning rate]*

### **Results and Discussion**

Iteration Results:

* Trial 1 ("Harry Style" and "PrinceTest1"):
  + Initial loss: 2112.54
  + Final loss (Iteration 4000): 441.25



* Trial 2 ("Try 2 Selena" and "Test2"):
  + Initial loss: 789.93
  + Final loss (Iteration 4000): 376.63



Visual Results:

* The final images at iteration 4000 for both trials exhibit distinct artistic transformations. Trial 1 presents a more dramatic style infusion compared to the subtler effect seen in Trial 2.

Analysis:

* Both trials demonstrated a consistent decrease in loss, indicating effective learning and adaptation of the style features.
* Trial 1 started with a higher initial loss, suggesting a more complex style transfer task, yet achieved impressive visual outcomes.
* Trial 2 had a lower initial and final loss, indicating a smoother convergence, possibly due to simpler style features or better initial alignment with the content image.

Challenges and Solutions:

* Challenge: Balancing style and content loss to avoid overpowering of either aspect.
  + Solution: Fine-tuning the weight parameters for style, content, and total variation loss.
* Challenge: Ensuring computational efficiency with high-resolution images.
  + Solution: Experimenting with image resizing and optimizing iteration counts.

### **Conclusion**

Summary:

* The experiments successfully demonstrated the effectiveness of neural style transfer using TensorFlow and VGG19.
* Both trials showed significant loss reduction, with Trial 1 handling a more complex style transfer and Trial 2 achieving smoother convergence.
* The final images highlighted the model's capability to blend styles and content, resulting in unique artistic creations.

Future Work:

* Experiment with different layer configurations in VGG19 to explore various style and content feature extractions.
* Test with diverse image styles and contents to assess the model's versatility.
* Investigate the impact of varying weight parameters on the balance between style and content in the output images.
* Explore real-time style transfer applications or video style transfer for dynamic content.
* Two minute (short): https://youtu.be/KYBCZ-9NyEU
* 15 minutes (long): https://youtu.be/uTNOB3NB\_IM