# Multi-Style Neural Transfer: Applying Diverse Artistic Styles to Image Quadrants

Rushitha Buddharaju, Srilatha Korrapati

July 24, 2024

## 1 Introduction

However, neural style transfer techniques have shifted waves of computational creativity since they have enabled the application of styles to content images. This evolution has urged art creation within digital media, graphics, and visualization of artworks. However, most introduced techniques work to change the entire image by employing a single artistic style. While such functionality is stimulating, the derived artistic approximations' degree of freedom and adaptability remain restrictively narrow. Our research explores a novel approach: multi-style transfer that allows multiple regions within an image to be transformed in a single process.

This combined approach directly answers two key research questions. Firstly, we examine how multiple style transfers can be applied to various regions of an image such that the overall stylized image retains reasonable compatibility with the original image and an appealing artistic look. This is one of a kind not only in the sense of the techniques involved in performing several style transfers but also the artistic one in combining different styles as part of a single artwork. Second, studying the semiotic and aesthetic consequences of applying multiple styles to a single picture is essential in the style transfer domain. This can include, for example, experimentation with a range of artistic forms that can be implemented to get better depersonalized results, imitating layering from human artists.

The rationale for our study is its deviation from the traditional approach and desire to expand the choice of works with greater philosophical depth. Combining many varieties into a single picture will allow the mimicking of singularities of the structures created by people, which often include various techniques, inspirations, and styles. It offers a direction to escalate neural style transfer to new heights in every aspect, making it a valuable asset for artists, designers, and content creators. From a technical perspective, it is also possible to note that this study can respond to the problem of creating an algorithm that can filter, modulate, and balance multiple style inputs pending in neural networks and image engineering.

## 2 Related Work

The exploration of neural style transfer has presented numerous improvements since it was first coined. Gatys et al. [1] initiated the idea in 2015 by illustrating that convolutional neural networks can disentangle and recombine the content and style of images of one's choice. The authors' method, based on the feature representations of deep networks, offered a new way of applying AI to creative tasks and attracted much attention in the scientific community.

Subsequently, Johnson et al.[2] proposed a method of real-time style transfer in 2016 based on Gatys' work. Their approach lessened the computational cost and time of the original approach by training a feed-forward network to solve Gatys et al.'s optimization equation. This invention made style transfer possible on platforms incorporated into consumer-level and mobile applications.

Li et al. [3] used the generic style transfer in 2017, which stands for a new approach whose focus is not on the styles learned by the model during training. Their method referred to the feature transforms that bring the content and style features statistics, offering more versatility in style application. This work was valuable in advancing the status of style transfer technology that can generate more styles without undergoing training.

A lot of work has been done in improving NTS. However, the methods developed are primarily geared at applying a single style to the entire image. Based on these works, our work further develops and proposes the new thought of using several styles in an image with the fixed quadrant method. This approach is similar to on-based approaches, which include the semantic style transfer described by Zhang et al. [4]. Still, this approach does not use semantic understanding of the image but rather geometric region obtaining. The end aim is to investigate this relation and synthesize a new approach toward generating new artistic styles under the neural style transfer model.

# 3 Dataset

This research used the "Image Style Transfer" dataset from Kaggle (Accessed from Kaggle: (https://www.kaggle.com/datasets/duttasd28/image-style-transfergoogle-images). This dataset can be used in our project since it includes only the content and style images of high selectiveness, allowing for more concrete experiments with different style combinations. The dataset's structure, which has two different directories – the content and style images – corresponds to our multi-style transfer approach.

The dataset is organized into two main directories: To be specific, "train," which has roughly 11 content images, and "art," which has around eight style images. In most cases, these images are in JPEG format, and the average dimensions of the images are 512 x 512. The benefits of this resolution include reasonable computational resource usage and sufficient image quality for our neural network processing. But before feeding the images to the neural style transfer model, the images undergo the following preprocessing steps. First, all images are resized to be of the exact dimensions of $256 \times 256$ pixels. This resizing makes it possible to compare inputs presented to the neural network and also assists in the use of resources. After resizing, the pixel values are normalized to the range [0, 1] and then are standardized by parameters Mean from the ImageNet dataset equal to [0. 485, 0. 456, 0. 406] and Std from the ImageNet dataset equal [0. 229, 0. 224, 0. 225]. This standardization is essential because this project uses a VGG19 pre-trained network for style transfer and standardizes the input data for a similar distribution as the pre-trained network.

To improve the stability of the proposed method, data enhancement is performed on content images. Such includes random flips along the horizontal axis and rotations by a

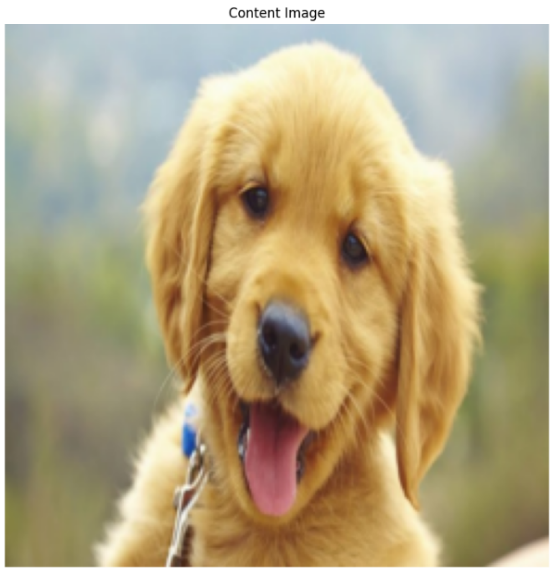



Figure 1: content image Input

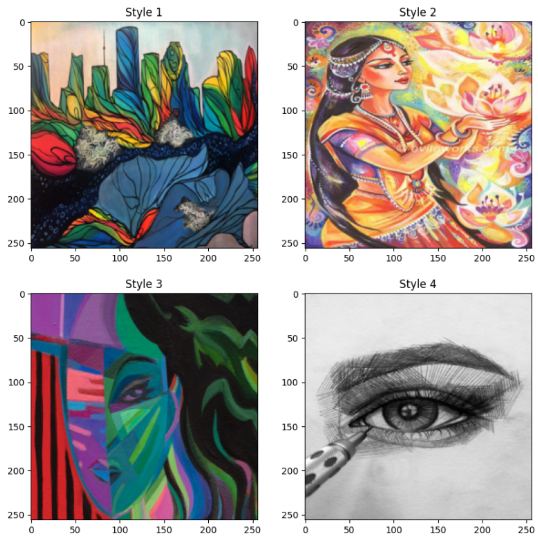



Figure 2: Style Images Input

certain angle within a given limit of 10 degrees. These augmentations enable our model to generalize better regarding different orientations and compositions of the images and reduce overfitting.

A critical process of our preprocessing pipeline is content image division into four quadrants. Before applying normalization, exactly like the previous step, each content image is split into four parts or quadrants. This division is the basis of segmentation in a multiple-style approach, thus allowing different styles to be used in various image regions. This division is done programmatically, so the same is done on all content images, enabling us to process each quadrant independently in our proposed style transfer algorithm.

# 4 Approach

Our method expands the NST introduced by Gatys et al. [1] deeply by introducing adaptations meant to enable the application of several styles on a single input image. We use a pre-trained VGG19 network for extracting content and style features; this was initially trained for image classification on the ImageNet database [5]. For content representation, we use the 'conv4_2' layer, which is powerful enough to capture the structure of the input image. Style representation involves multiple layers: 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1', and 'conv5_1' Streams. It permits us to have layers that contain stylistic information of different scales ranging from fine scales to big stylistic scales.

The transfer process starts with the pre-processing of the input images. Four style images belong to the content image, and their analysis depends on dividing the content image into four equal quadrants. We then extract content features for each content quadrant and style features for each style image related to that quadrant. The features of the style are employed to calculate the Gram matrices that quantify the correlation of the feature maps and thus reflect the style of the image.

The key that defines our method is the step of optimization. For each quadrant, we build a target image that begins with the content of the particular quadrant. We then repeatedly update this target image to minimize both the content loss and the style loss. Content loss is the squared difference between the target's feature values and the image's content. Style loss is defined as the sum of squares of the Euclidean distance of the Gram matrices of the target and style images. The idea that opalescent text is based on is of losing once twice – once for structure and, for the second time, for stylistic peculiarities, which we want to implement.

## 4.1 Mathematical Formulation

The loss functions used in NST are crucial for optimizing the artistic style while preserving the content of the image.

### 4.1.1 Content Loss

The content loss measures the difference between the content of the generated image and the content image. The content loss for a given layer $l$ can be expressed as:

$$L^l_{\text{content}} = \frac{1}{2} \sum_{i,j} (F^l_{i,j} - P^l_{i,j})^2 \tag{1}$$

where $F^l$ and $P^l$ are the feature maps of the generated and content images, respectively, at layer $l$.

### 4.1.2 Style Loss

The style loss measures the difference between the style of the generated image and the style image. The style loss is computed using the Gram matrix, $G^l$, of the feature maps. For a layer $l$:

$$L^l_{\text{style}} = \frac{1}{4(N_l^2 M_l^2)} \sum_{i,j} (G^l_{i,j} - A^l_{i,j})^2 \tag{2}$$

where $G^l$ is the Gram matrix of the generated image and $A^l$ is the Gram matrix of the style image. $N_l$ and $M_l$ are the dimensions of the feature map.

#### 4.1.3 Total Loss

The total loss is a weighted sum of the content loss and style loss:

$$L_{\text{total}} = \alpha L_{\text{content}} + \beta \sum_{l} L^{l}_{\text{style}} \tag{3}$$

where $\alpha$ and $\beta$ are weights that balance the content and style contributions. $L_{\text{content}}$ is typically computed for one layer, while $L_{\text{style}}$ is computed across multiple layers.

We use the Adam optimizer with the learning rate set at 0.003 and perform 5000 iterations for each of the quadrants to get the efficient implementation of style. Thus, the balance between content preservation and style application is maintained through weighting factors used in the content and style losses. These weights are crucial in determining how much of the originality of the content has to be maintained and how strictly the stylistic approach would be followed.

```
Step 500: Total loss: 195044229251072.0
Step 1000: Total loss: 91403438784512.0
Step 1500: Total loss: 57190396198912.0
Step 2000: Total loss: 42309479563264.0
Step 2500: Total loss: 33835752357888.0
Step 3000: Total loss: 28129930248192.0
Step 3500: Total loss: 23922846203904.0
Step 4000: Total loss: 20632376967168.0
Step 4500: Total loss: 18031310798848.0
Step 5000: Total loss: 15910278529024.0
Step 500: Total loss: 178542679687168.0
Step 1000: Total loss: 69236022050816.0
Step 1500: Total loss: 43655565934592.0
Step 2000: Total loss: 32560646193152.0
Step 2500: Total loss: 26187768266752.0
Step 3000: Total loss: 21987233628160.0
Step 3500: Total loss: 18949668339712.0
Step 4000: Total loss: 16654429847552.0
Step 4500: Total loss: 14909778690048.0
Step 5000: Total loss: 13495083991040.0
Step 500: Total loss: 190424824152064.0
Step 1000: Total loss: 77274388889600.0
Step 1500: Total loss: 42132253442048.0
Step 2000: Total loss: 31809148551168.0
Step 2500: Total loss: 25931817156608.0
Step 3000: Total loss: 21825549500416.0
Step 3500: Total loss: 18844259188736.0
Step 4000: Total loss: 16598011215872.0
Step 4500: Total loss: 14865969184768.0
Step 5000: Total loss: 13514549755904.0
Step 500: Total loss: 35904945127424.0
Step 1000: Total loss: 24199198408704.0
Step 1500: Total loss: 19620838768640.0
Step 2000: Total loss: 16097724071936.0
Step 2500: Total loss: 13033313140736.0
Step 3000: Total loss: 10596051320832.0
Step 3500: Total loss: 8875198644224.0
Step 4000: Total loss: 7771902705664.0
Step 4500: Total loss: 7054341177344.0
Step 5000: Total loss: 6553295388672.0
```

Figure 3: Iterations for each of the quadrants

Once all the quadrants are optimized, the styled quadrants are arranged in their previous position to get the whole picture. This reconstruction step is crucial for joining the processed regions of the image, which is divided into four quadrants, into one final result.

These experiments have shown a many-to-many relationship between the differently styled quadrants. Interpreting different styles in one picture yields more spectacular results than a single-style transfer approach. However, we've also experienced difficulties regarding the global coordination of the image, especially between different quadrants.

Future work is needed in several areas: We hope to subsequently adjust many of the model's hyper parameters with emphasis on the content and the style parameters of each of the four quadrants' positions. This fine-tuning is essential to make the stylistic transitions as unobtrusive as possible in the overall process of narration and, at the same time, to retain the content's communicative function.

# 5 Results

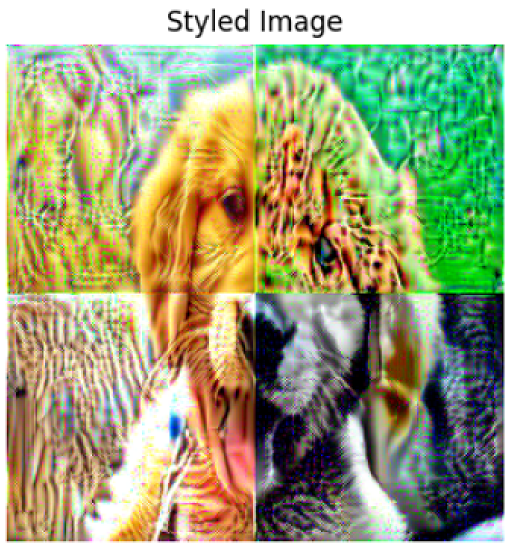
In the case of images, we applied our multi-style neural transfer method to a photograph of a Golden Retriever puppy. The original image depicted a young, golden-colored dog with large, expressive eyes and an open mouth, set against a blurred natural background.

Our quadrant-based style transfer technique dramatically transformed the image's color palette, textures, and overall aesthetic.

Each quadrant of the resulting image showcases a distinct artistic style:

Top-left: Dominated by cool blues and aquamarine hues, this quadrant exhibits a fluid, water-like pattern. Top-right: Vibrant greens and a mosaic-like texture characterize this quadrant. Bottom-left: Featuring a blend of cool blues and purples, this section maintains the fluid quality but with a more abstract appearance. Bottom-right: While retaining the original color scheme, this quadrant showcases enhanced texture and detail. width=0.51.png

[Result1]




[Result2]

Figure 4: Initial Results

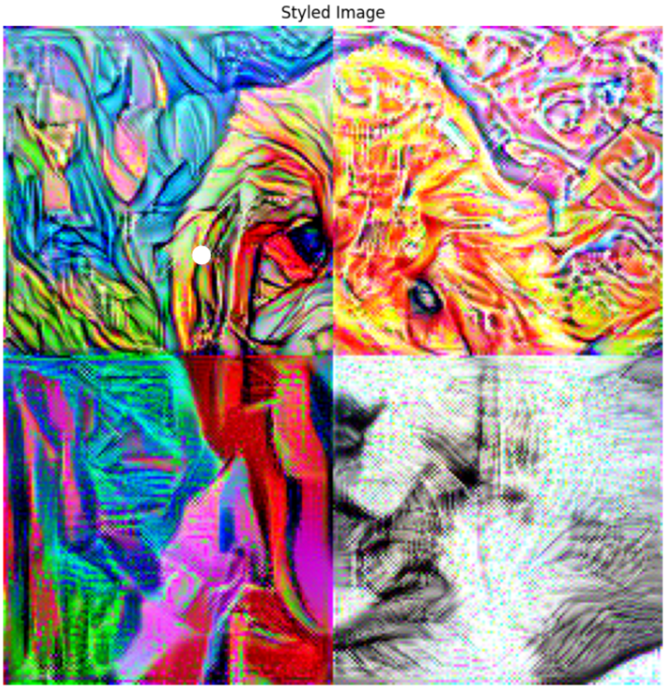



Figure 5: Final Styled Image

Remarkably, despite the diverse artistic styles applied to each quadrant, the puppy's identity and expression remain easily recognizable throughout the image. This outcome demonstrates our algorithm's ability to effectively blend distinct styles while preserving the core content.

The results of this experiment highlight the potential of our proposed approach to create sophisticated, multi-layered artistic interpretations while maintaining the integrity of the original subject matter.

# 6 Computational Considerations

The computational efficiency of neural style transfer can be significantly enhanced using GPUs compared to CPUs. GPUs offer parallel processing capabilities, allowing for faster execution of the numerous matrix operations involved in style transfer. This speed-up is crucial for practical applications, enabling real-time or near-real-time style transfer and reducing the time required for model training and optimization. In our experiments, using a GPU reduced processing time considerably compared to a CPU, making it an essential component for handling the computational demands of multi-style transfer.

# 7 References

# References


[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[2] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, Springer, 2016, pp. 694-711.

[3] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *Advances in Neural Information Processing Systems*, 2017, pp. 386-396.

[4] Y. Li, Z. Wang, J. Yang, C. Fang, and M.-H. Yang, "Universal style transfer via feature transforms," in *Advances in Neural Information Processing Systems*, 2017, pp. 386-396.

[5] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151-7160.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.