10-701 Project Report: Multi-Style Transfer

Sanjana Kuchibhotla, Shreya Singh, Sanjna Enjeeti, Harshdeep Kaur

December 11, 2024

Abstract

This paper addresses the challenge of multiple-style transfer, with the aim being to extend it beyond the existing single-style transfer methods. We developed a novel approach to solve this problem: the user can input words describing the styles they would like to transfer, and then an image is generated based off of the words. Our implementation uses the WikiArt dataset to capture characteristics from various art styles or art movements (e.g. Impressionism, Romanticism, etc.). We evaluated our methods by viewing the emphasis of the style on the content image. The goal was to ensure that the content image did not lose its visibility after the style was applied.

Experimental results across the different genres revealed that this method of searching and merging different art styles based on user input not only preserves the detail of the content, but it applies a version combining unique features from the included styles. Future work related to this project includes implementing weights to give some styles more emphasis over others and creating a recommendation system based on characteristics in the content image.

1 Introduction

Our goal is to approach the challenge of multi-style transfer, where the objective is to apply the styles of multiple images (rather than just one) to a single content image. Traditional style transfer involves using the style of one "style image" to the structure and context of a "content image." However, we want to expand on this concept by integrating stylistic elements from multiple style images, which would allow for generating content images that reflect a blend of various artistic styles.

This challenge is important to solve because it has significant value in scenarios such as capturing the essence of a specific artist's work (e.g. Van Gogh) or capturing the features that were essential to an entire art movement (e.g. Impressionism). By leveraging multiple styles simultaneously, we can create content that combines diverse influences, providing a more versatile visualization of artistic styles.

2 Summary of Data

Our data used images for both the content and style that we want to transfer. This is best suited for this problem since it uses the style and aesthetics of images to transfer onto other images' content, which is a visual problem. The dataset we used for style was the WikiArt dataset. This dataset was well suited for our use case since it included multiple artistic pictures as well as the artist, genre, and style (e.g. Realism) of each image.

Here is an example of an image in our dataset:

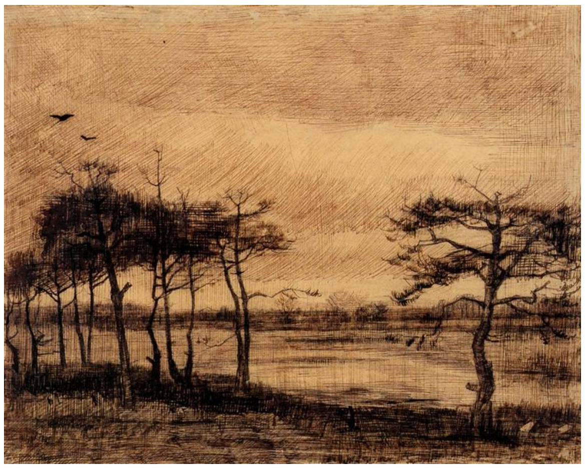

Figure 1: Image Example

This image has artist label `vincent-van-gogh`, genre label of ‘Landscape’, and style label ‘Realism’.

The dataset includes similar images from 24 different artists, 11 different genres, and 16 different styles, encompassing a variety of different images and aesthetics. This diversity in the dataset made it well suited for our use case, which includes not only style transfer but also “searching” for images that match specific queries based on stylistic features. For example, if a user searched for “impressionist style with dark colors,” we would be able to identify and retrieve images that correspond to these attributes. The selected images would then be utilized for multi-style transfer, enabling the application of a combination of stylistic influences to the target content image.

There is a small amount of bias that comes along with using this dataset. Certain styles, like Cubism, only have about 300 images associated with it, compared to 3000 images associated with Impressionism. This may lead to a worse depiction of images from the underrepresented categories. Other than this, though, the dataset is taken from WikiArt, which is a reputable source that contains quite a few pieces with accurate information. So, although this dataset contains some bias, there isn’t a significant amount.

3 Methods

One model we were thinking of using for our baseline was StyleGAN, which is a prominent state-of-the-art model in style transfer. However, StyleGAN is implemented only on one content image and one style image. We want to implement a use case that will be able to handle multiple style images with one content image. This will allow for more use cases - searching for a particular image style to apply to the content image, rather than specifically finding a style image that matches the exact style of what someone would want, for example.

We wanted to use StyleGAN as a baseline since StyleGAN’s generative adversarial network (GAN) architecture is designed to produce realistic images. StyleGAN first uses a mapping network to transform input latent vectors into an intermediate latent space, which then influences the generator at each convolutional layer through adaptive instance normalization. This setup helps separate big-picture features from smaller details and makes it easier to mix styles or smoothly blend images together. This also allows mixing and blending styles, which improves image quality and makes features more independent [3].

We ran into hardware limitations when using StyleGAN and instead used Neural Style Transfer as our baseline, transferring the style of one image into the content of another. We looked at ideas implemented in Neural Style Transfer and modified them to work for multiple images. While StyleGAN generates a new image based on multiple styles and synthesizes it based on a latent vector, Neural Style Transfer just transfers style onto a content image.

3.1 Neural Style Transfer

Neural Style Transfer utilizes the concept of "content loss" and "style loss". With just one style image and one content image, the idea is to take these two losses and minimize them. The content loss essentially measures the distance in content between the target image and the content image, while the style loss measures the distance in style between the style image and target image. Therefore, the overall loss is just a linear combination of the style and content losses:

Letting p be the content image, a be the style image, and x be the target image, we can define the overall loss function as follows:

$$\mathcal{L} = \alpha\mathcal{L}_c(p,x) + \beta\mathcal{L}_s(a,x)$$

where $\mathcal{L}_c$ is the content loss and $\mathcal{L}_s$ is the style loss.

Now, the content loss is simply just the squared error loss between the content and target images (where F is the feature representation of the content image and P is the feature representation of the target image, and l is the layer):

$$\mathcal{L}_c(p,x,l) = \frac{1}{2}\sum_{i.j}(F^l_{i,j} - P^l_{i,j})^2$$

The style loss is slightly more involved, using the concept of Gram matrices in order to get assign a numerical value for "style". The Gram matrix captures the correlations between different feature channels, and captures how strongly they interact with each other. It is calculated as:

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

Then as the Gram matrix considers each layer, this is just

$$G^l_{ij} = \sum_k F^l_{ik}F^l_{jk}$$

To use this in the style loss, we calculate the mean-squared distance between any two Gram matrices for every layer, and take the sum across all layers [1, 2]. We then used gradient descent with the Adam optimizer and a learning rate of 0.01 to optimize the total summed loss function (style + content).

$$\mathcal{L}_s(a,x) = \frac{1}{4N^2M^2}\sum_{i,j}(G_{ij} - G'_{ij})^2$$

Here, N is the number of features, and M is the number of elements in each feature. G is the gram matrix for the style image, and G' is the gram matrix for the generated image. $\frac{1}{4N^2M^2}$ is the normalizing constant here.

The baseline output of regular Neural Style Transfer between one content image and one style image for both the Post-Impressionism and Romanticism styles are shown below:

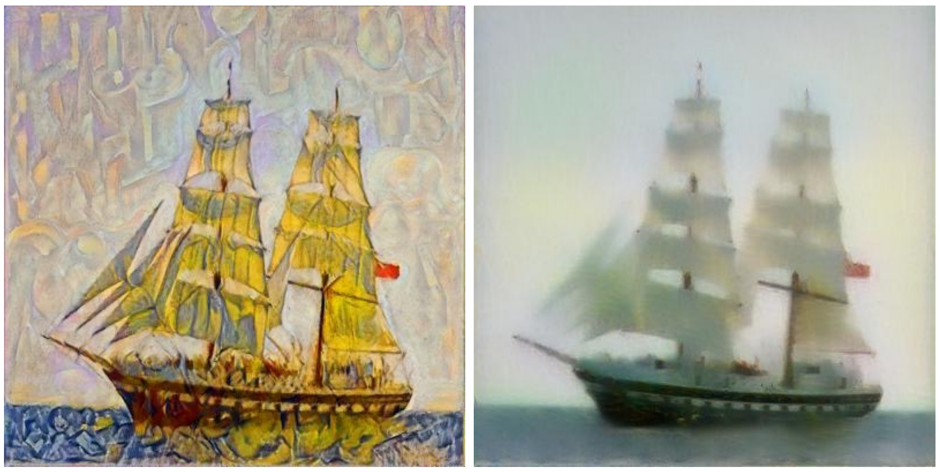
Figure 2: Neural Style Transfer Baseline

For our project, we attempted to modify the Neural Style Transfer algorithm to better utilize multiple images as input. We approached this in two ways: our first idea was to combine the losses from each style picture into one loss function and generate the image with this loss function minimized. Our second idea was to just combine all the style images into one general style picture, and proceed with normal Neural Style Transfer in order to generate the combined image.

3.2 Method 1: Combine Losses

Our first method of modifying this algorithm was to update the style loss. Currently, our algorithm computes the style loss between our generated image and each of our style images, then averages this result to get an overall style loss. It then weights the content loss and the style loss according to the parameters α and β. More formally, if we let $\mathcal{L}_{s,i}$ denote the style loss on the ith style image and let a_i denote the ith style image, our overall loss using this method is

$$\mathcal{L} = \alpha \mathcal{L}_c(p, x) + \frac{\beta}{N} \sum_{i=1}^{N} \mathcal{L}_{s,i}(a_i, x)$$

3.3 Method 2: Combine Style Images

Our second approach for implementing multi-style transfer was to combine multiple style images into a single style image and proceed with regular Neural Style Transfer. Rather than updating the loss function here, we generated a new image that was representative of all the styles in the many style images combined. Originally we wanted to use a model like StyleGAN to generate this new style image, but we were running into hardware issues, and ended up just creating a new image by layering the style images on top of each other.

3.4 Cosine Similarity Search

One main feature we wanted to implement was using cosine similarity search for different styles. This included inputting some kind of search query as well as an image, and outputting the content image that most reflected the style of the original query. In order to do this, we used cosine similarity search to find the most similar embedding between the user's input and the different values of the features in the dataset (genre, style, artist).

4 Results

We tested our model on both methods in order to see how each method performed on different styles. We tested the following image obtained from the ImageNet dataset:

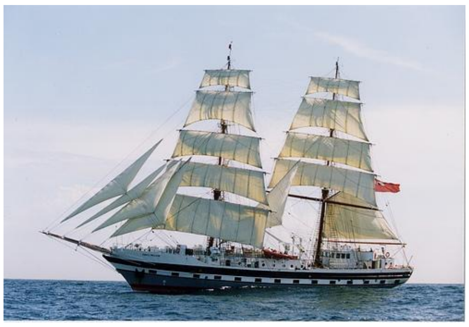

Figure 3: Content Image

We first used similarity search with the original query to test what style was most similar to our query - we ended up with the following similarities for the query “Atmospheric”:

```
{'Abstract_Expressionism': 0.66780704,
'Action_painting': 0.578887,
'Analytical_Cubism': 0.5905356,
'Art_Nouveau': 0.5650048,
'Baroque': 0.644004,
'Color_Field_Painting': 0.54468596,
'Contemporary_Realism': 0.64140946,
'Cubism': 0.5926007,
'Early_Renaissance': 0.6880181,
'Expressionism': 0.63165885,
'Fauvism': 0.66558754,
'High_Renaissance': 0.68823063,
'Impressionism': 0.6943008,
'Mannerism_Late_Renaissance': 0.5872085,
'Minimalism': 0.64198995,
'Naive_Art_Primitivism': 0.6158701,
'New_Realism': 0.6912294,
'Northern_Renaissance': 0.6484735,
'Pointillism': 0.66965365,
'Pop_Art': 0.59362936,
'Post_Impressionism': 0.7144661,
'Realism': 0.704267,
'Rococo': 0.6545713,
'Romanticism': 0.6514977,
'Symbolism': 0.7072947,
'Synthetic_Cubism': 0.60005957,
'Ukiyo_e': 0.6415151}
```

As the query most closely matched with the 'Post-Impressionism' style, we proceed to use style transfer with all the images corresponding to the 'Post-Impressionism' style.

Here are some examples of the pictures in the "Post-Impressionism" style:

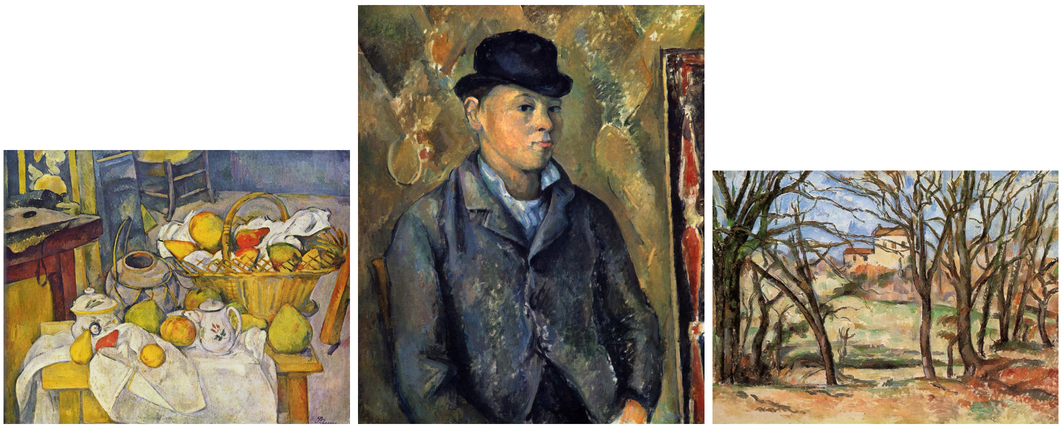

Figure 4: Post Impressionism Figures

We also tested our similarity search on each method with the new query "love" which directed us to the "Romanticism" style of images.

Here are some examples of the pictures in the "Romanticism" style:

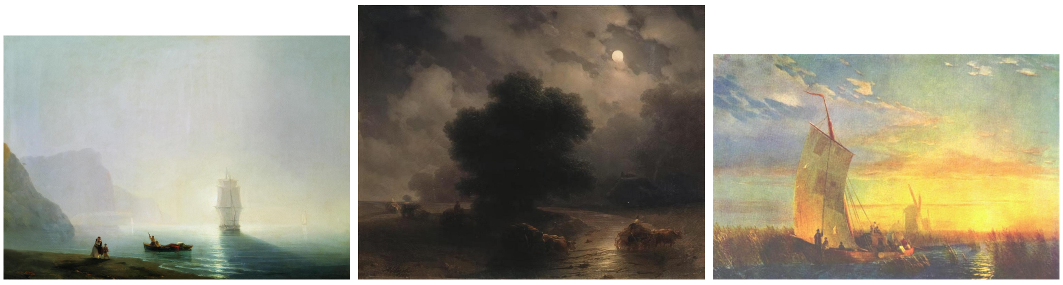

Figure 5: Romanticism Figures

4.1 Method 1: Combine Losses with Similarity Search

4.1.1 Post-Impressionism Results

We first tested the combining losses method on the Content Image (Fig. 3). We trained it for 1000 epochs with a learning rate of 0.01. The loss plot is shown below:

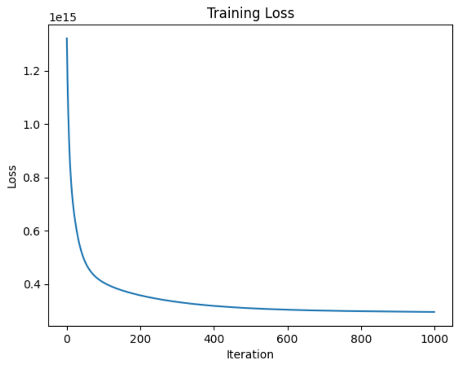

Figure 6: Post-Impressionism Multi-Loss Loss Plot

Here, we can see how the loss decreased smoothly over all the epochs. The loss is quite high, but this is due to the nature of our loss function – we've summed up and averaged the losses on various images, leading to a relatively higher loss.

The final outputted image is shown below:

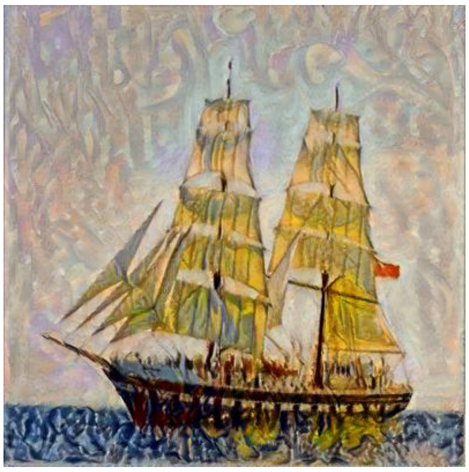

Figure 7: Post-Impressionism Multi-Loss Style Transfer

This picture shows the final image with style transferred with the original search query "Atmospheric". We qualitatively analyzed the output image, comparing it to our baseline image and other results. We can clearly see the influence of the 'Post-Impressionism' style images applied to the content image here. The sails especially show similar strokes as in the original Post Impressionism images, and even the water has a modified brushstroke.

Comparing this with the baseline, we can see more detail compared to a pretty blurry image with the baseline. Specifically, the base of the ship in the multi-style image is much more detailed and resembles the original content image more closely that the original Neural Style Transfer baseline, in which there is less detail. In addition, the brushstrokes in the sky are less distracting and seem more natural and nuanced.

4.1.2 Romanticism Results

With the query "love", we generated the Romanticism style as the most similar style to our query. We then tested the output of Multi-Loss Style Transfer using the combined losses on the original content image.

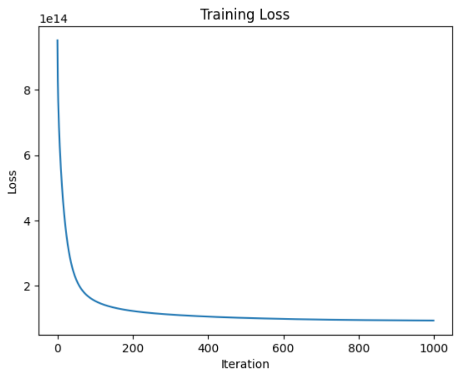

Figure 8: Romanticism Multi-Loss Loss Plot

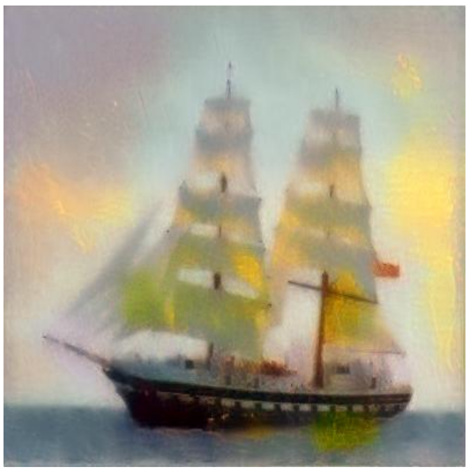

Figure 9: Romanticism Multi-Loss Style Transfer

This image shows the results of running method 1 (the multiple averaged loss method) on our ship image in the style of Romanticism. We can see the influence of the Romanticism style in this image – there are softer brush strokes than in the Post-Impressionist paintings and when compared to the original image. Additionally, this method outperformed the baseline by taking into account more of the features of the style of Romanticism. It kept more of the contrast as was present in the original image while also taking influence from all of the romantic era paintings we provided as input. This method also resulted in a bit brighter output than method 2.

4.2 Method 2: Combine Style Images with Similarity Search

4.2.1 Post-Impressionism Results

We again used a subset of the Post-Impressionism images in order to output our style image for this method. Here is the combined layered style image used for this method:

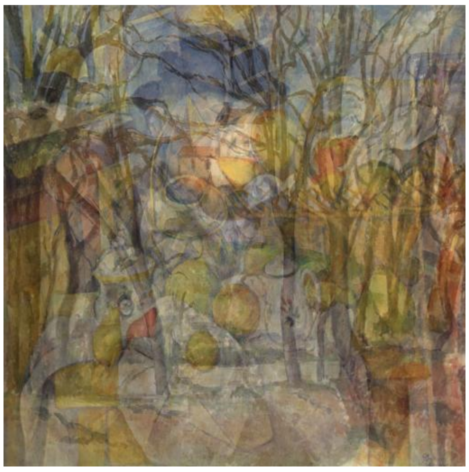

Figure 10: Post-Impressionism Layered Style Image

We again tested this method on the Content Image (Fig. 3) with the combined style image for 1000 epochs with a learning rate of 0.01. The loss function for this method is shown below:

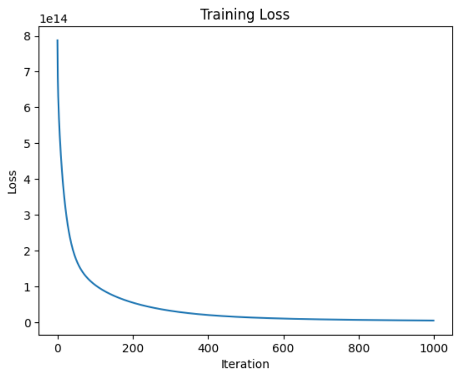

Figure 11: Post-Impressionism Layered Loss Plot

Again, we see a smooth loss curve, with a lower loss than the Multi-Loss method. However, this is due to the fact that we are not summing up the losses in this method.

The output image for the Post-Impressionism style with the query "Atmospheric" is shown below:

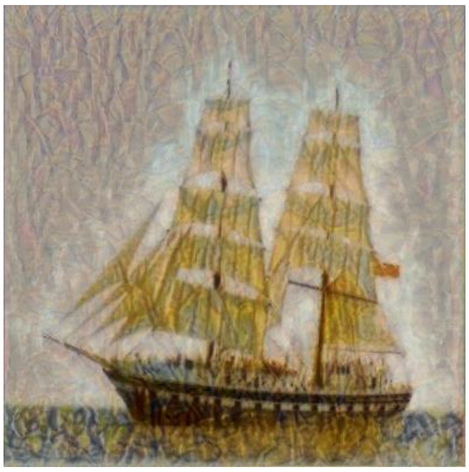

Figure 12: Post-Impressionism Layered Style Transfer

This method performed decently well, preserving the content of the original image and applying the style in some manner to the content image. However, there is a lot more noise than in the combined loss method, and features in the image such as the base of the ship and the sails seem less clear. In addition, the sky seems less naturally painted and a little more noisy than compared to the combined loss method. The image also seems to have much darker undertones than both the baseline image and the combined loss function image, showing that maybe combining the images led to a more biased result.

4.2.2 Romanticism Results

For the Romanticism style with the layering method, we generated our combined style image from the subset of three images from the Romanticism style in the dataset. This was the combined image used:

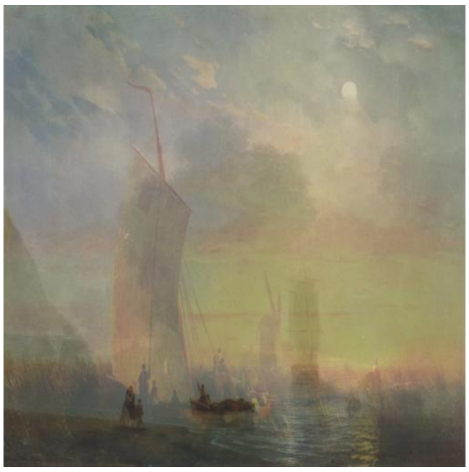

Figure 13: Romanticism Layered Style Image

Similar to the layered combined image from the Post-Impressionism style, we can see influence from all the different style images in the combined image. However, since they are layered on top of each other, some dark undertones are added which contributes to the undertone of the final output image produced.

The loss plot for the content image with the layered style image is shown below. It again has a smooth curve but the loss starts off quite high.

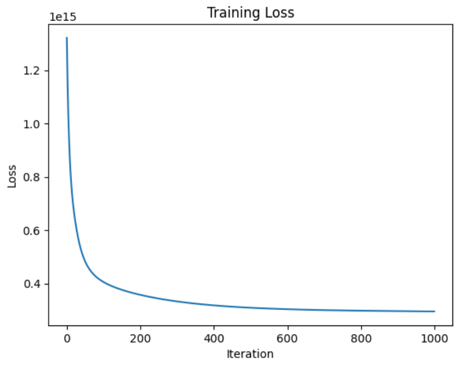

Figure 14: Romanticism Layered Loss Plot

The final output image for the layered image style transfer onto the original content image is shown below:

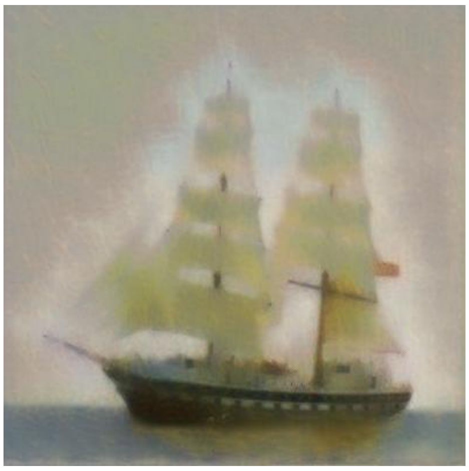

Figure 15: Romanticism Layered Style Transfer

While this performed well in terms of projecting the style of all the style images onto the content image, we can also see that there are a lot of preserved dark undertones in the final output image. It also appears more dull than the Multi-Loss version as well as the baseline image. The style seems to be more representative of the first and second romanticism images (see Fig. 6) than the third image, while all three images seem represented more in the Multi-Loss version.

5 Discussion and Analysis

The method we used was appropriate for the challenge we were attempted to solve. We were attempting to perform style transfer more generally, and the methods we used reflected that. We used the baseline method of Neural Style Transfer and improved it in order to work for multiple style images. We tested two different experiments for style transfer and qualitatively evaluated the results. We got decent results with our cosine similarity search, but it was more difficult to determine differences between our experiments otherwise. We aren't able to compare losses here, as the loss function varies from method to method, so we had to analyze our results qualitatively. The combining losses method seemed to perform better on both the Post-Impressionism style images as well as the Romanticism style images. This was seen through the brightness of the output image as well as the fact that it seemed to place more of an equal influence on each image rather than being biased towards a few images. However, this result is not decisive, as we were not able to analyze all the images from the dataset corresponding to a particular style/genre/artist. Given more time and resources, we would want to expand our testing so that it incorporated a larger and more diverse variety of images. In addition, generating a combined image that is more reflective of each image could work better for style transfer, rather than the layering method, which is something we would want to test in the future.

There were multiple limitations that we ran into while trying to implement the project. Our biggest limitation was our lack of compute power, resulting in a limit on the ideas we could use as well as on the amount of data we were able to process at a time. Because of this, we were not fully able to analyze our multi-style transfer on all the images corresponding to a particular style/genre/artist, but rather could only analyze it on a subset of the given style/genre/artist. If we had more time and more resources, we would try running this on many more images in order to see if it performs better in any way. Overall, our results were not necessarily conclusive, but given more time and compute power we would hope to get more decisive results.

Another limitation we ran into was the ability to use StyleGAN. We wanted to use StyleGAN as a baseline model and wanted to build our multi-style transfer on top of StyleGAN, but we ran into hardware limitations on what we were able to use. An improvement and further goal for our project if we had more resources and time would also be to try using StyleGAN as a baseline.

Future researchers can see if there is any way to improve the loss function given multiple images (rather than just adding the losses from each transfer). We could also consider weighting different images differently, potentially depending on how representative a given image is of a certain style. Another possibility for improvement could be to see if there is another way to combine the style images into a new image that is not just the original n images layered. This was an idea we had originally, but were unable to fully execute due to the limited compute power that we had.

References

[1] Aldo Ferlatti. Neural style transfer (nst): Theory and implementation, 2023. Accessed: 2024-11-20.

[2] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.