Neural Style Transfer

Nathan Warren
Interdisciplinary Data Science
Duke University
Naw32@duke.edu

Abstract

Neural Style Transfer is process of incorporating the semantic content information from one image and the style features of another image to create a hybrid image. The goal of this project was to recreate Gatys *et al.*'s Neural Style Transfer algorithm using VGG-19 and test how images of different complexity for content and style contributed to a hybrid image. Proper recreation of Gatys *et al.*'s was achieved and it was found that 'freeform' abstract style artworks led to less boundaries from initial content images being kept. Using less abstract style art pieces created images that were generally more aesthetically pleasing. Furthermore, it was shown that initialization image played a large role in the final output image. When initialized with style or content images, more style or content would be apparent in the final image. When initializing with random noise, the resulting hybrid image was less comprehensible. These results provide insight into further areas to be explored in the Neural Transfer Style space.

1 Background

Convolutional Neural Networks (CNNs) are primarily used for object detection due to their ability to capture and learn lower and higher level features of an object of interest. However, thanks to growing interest and development in the deep learning field, investigation into use cases outside of classification have been shown to be quite fruitful. One such application is the ability to generate works of art.

Over the past two decades many algorithms have been developed to generate art through texture synthesis. The goal of many of these algorithms is to transfer the texture of an image to any other image such that it would be perceived by humans as the same texture [1]. This is done by either replacing pixels in the original seed image one at a time or in patches, which is known as image quilting, while maintaining the higher level features of the seed image, which are defined as shapes or the spatial arrangement of the seed image[1][2]. While these algorithms are able to achieve texture/style transfer, they only have the ability to transfer low-level features, such as color, from the style image to the seed image [3]. To generate a more efficient style transfer that was capable of extracting the semantic content from a style image, Gatys *et al.*, developed a style transfer method using a CNN.

The core idea of Gatys *et al.*'s *Neural Algorithm of Artistic Style*, is that well-performing CNNs, which have already learned features, can be manipulated to extract representations of content and style independently from photos and works of art [4]. Unlike previous algorithms, this algorithm is able to extract both low level features pertaining to style and higher level features, such as objects and scenery, from content images. Since these representations are separable, they can then be used to create a hybrid image from content and style images. This process is called Neural Style Transfer (NST).

ECE 590-03 (2020 Fall) final report.

The goal of this paper is to replicate Gatys *et al.*, NST. Original images used by Gatys *et al.*, will be used to compare how faithful the reproduction of hybrid images is and new images will also be used to gain an understanding of where the algorithm may falter.

2 Methods

2.1 Preprocessing

In order to work with an image in PyTorch, the style and content image must go through resizing in order for the convolutional neural network to understand the images. The images are not directly input into the model but rather must be converted to tensors and a batch dimension of 1 must be added since a single image will be processed at a time. Images are normalized using the mean and standard deviation RGB values of ImageNet as VGG-19 was originally pre-trained on ImageNet. This is done to reduce internal covariate shift as it ensures that all images regardless of original pixel values have a similar distribution so that features are able to be detected in a robust manner. Resizing is done as the images, which become tensors, need to be the same square dimensions going into the network. A pixel size of 512x512 was used although any dimensions could be used. The higher the number of pixels without overshooting the original size of the image, the higher the quality of the output image. Lastly, an image/tensor must be created that will become the hybrid output image of the model. Random noise, style or content images can be used, although naturally, the starting image will influence the spatial features of the output image after being run though the model. The output image must also be set to "requires grad" in PyTorch so that the pixel values will be modified to optimize the loss. To avoid confusion, the starting image will be referred to as the output image regardless if it has been run through the model or not.

2.2 Content and Style Layers

For simplicity of layer labeling, here on forward, layers will be named according to their position relative to a pooling layer that came previous to a given layer. 1-1 would indicate the first layer of the model, while 2-2 would indicate the second convolutional layer after the first pooling operation.

The only content feature map that was used was layer 4-2. This layer was chosen as layers that are later in the model generally extract higher level features such as shape and global arrangement of objects. Therefore, these features can be seen as describing more of the “content” of the image than earlier layers.

The following style layers were used: 1-1, 2-1, 3-1, 4-1, and 5-1. These layers were chosen as layers earlier in the model generally contain more features regarding the color of the image, thereby allowing for near perfect pixel value extraction from the original image. Earlier layers also tend to capture more texture-like features. Both color and texture can be thought of as describing the “style” of an image.

2.3 Loss Function

In order to combine the two images a loss function must be created that incorporates how different the output image is from the content image and the style image. When the network optimizes the loss, by changing the pixel values, it will generate an image that is representative of a combination of the style image and content image.

Content Loss

By individually running our content and our output tensors through the model, we can obtain their feature maps at 4-2 using forward hooks as previously mentioned. Let $\vec{p}$ and $\vec{x}$ be the original content image and generated output image. Their respective feature maps at layer *l* are P^l and F^l. We can define our content loss as the mean square error between the original content image feature maps ($P^l_{i,j}$), and the generated output image feature maps ($F^l_{i,j}$) at layer *l*.

$$\mathcal{L}_{\text{content}(\vec{p},\vec{x},l)} = \frac{1}{2}\sum_{i,j}(F^l_{ij} - P^l_{ij})^2$$

Style Loss

Similar to the content feature map extraction, feature maps for style are generated by individually running the original style and output tensor through the modified VGG-19 network. However, since multiple feature maps are being extracted, a gramian matrix is used to take the inner product of these feature maps. The height and width of the feature maps from the style layer are multiplied together and then the inner product, G^l_{ij} is found which can be thought of as a correlation between feature maps, $F^l_{i,j}$ in layer *l*.

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

The inner products are then normalized by the number of pixels in the style layer, *l*. The style loss (E_l) can then be defined as the mean square error between the original image, A^l_{ij} and the generated output image G^l_{ij}. Let $\vec{a}$ and $\vec{x}$ be defined as the original style image and the generated output image and let A^l and G^l be their style feature maps at layer l.

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G^l_{ij} - A^l_{ij})^2$$

Once calculated for both the output feature maps and the style feature maps, weighting can be applied to these style layer normalized inner products to influence the strength of different style layers on the overall style of the image.

$$\mathcal{L}_{\text{style}}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$$

Using the style loss and the content loss, we can define our total loss with a simple addition of the two. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are hyperparameters which can be used to tune the loss thereby influencing the total representation of style and content of the final output image.

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \boldsymbol{\alpha}\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \boldsymbol{\beta}\mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

2.4 Model

VGG-19, which was pre-trained on ImageNet was used. Only the "feature" layers were used meaning that the "classifier" segment of the model, which is comprised of the three final linear layers, were removed. The model was set in "eval mode" so that weights would not be altered when images ran through it. All max pooling layers were changed to average pooling layers with a kernel size of 2 and a stride of 2. Forward hooks were used to extract content, style, and output feature maps at specified layers. Total loss was calculated, as previously mentioned, and the gradient was used to update the pixel values to optimize the loss. LBFGS was used as the optimizer as it proved to generate hybrid images quicker than ADAM. The output image ran through the model for 30 iterations and the final output image was un-normalized and saved.

3 Experiment results

3.1 Comparison with Gatys *et al.*'s NST

To properly compare to Gatys work, Tuebingen Neckarfront was used as the content image. *The Shipwreck* and *Starry Night* were used as style images for an exact comparison. All hybrid output images were generated using an $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ of 1 and 1e4, respectively. Each output image passed through 30 iterations and was initialized with the content image. Style weights were set to 1e3/64, 1e3/128, 1e3/256, 1e3/512, and 1e3/512 to replicate Gatys *et al.*'s work.

As can be seen in Figure 2, style transfer did occur but differently than in Gatys *et al.*'s work. Gatys *et al.*'s work features more distortion of buildings and has more colors from the style. The difference here may be attributed to the initialization image as in Gaty's work, the initialization image was white noise. Gaty's work also appears to be a bit darker than my attempted recreation but I believe this

Figure 1: Tuebingen Neckarfront will be used as the content image.

may be due to how images were normalized. Overall, the blue and yellow colors and texture of brush strokes from Van Gogh's *Starry Night* can be see in my recreation; therefore style transfer was a success.

Figure 2: Comparison between Gatys *et al.*'s NST (left column) and my attempted replication (right column). The two style images used were *Starry Night* by Vincent Van Gogh, and *The Shipwreck* by J. M. W. Turner and William Turner. Original style images used are shown in the bottom left corner of their respective rows. Style transfer was successfully achieved although not to the same extent as in Gatys *et al.*'s work.

3.2 Testing Different Image Complexities *et al.*'s NST

A picture of the Duke Chapel and a picture of rocks were used to test complexity. The picture of the Duke Chapel is more complex than Tuebingen Neckarfront as it has more clouds and trees which are quite texture detailed. The image of rocks is meant to be a simple image as most rocks are uniform in color and all rocks in the photo are about the same distance from the camera. The style images

also vary in complexity. Jackson Pollock's *Number One*, painting is extremely complex using many colors and lines. Van Gogh's *Irises* has a very limited assortment of colors but many objects, and Kandinsky's *Composition VII* is again abstract and contains a multitude of colors and shapes. The same hyperparameters used in 3.1 were used to test how the content and style complexities impacted the output image.

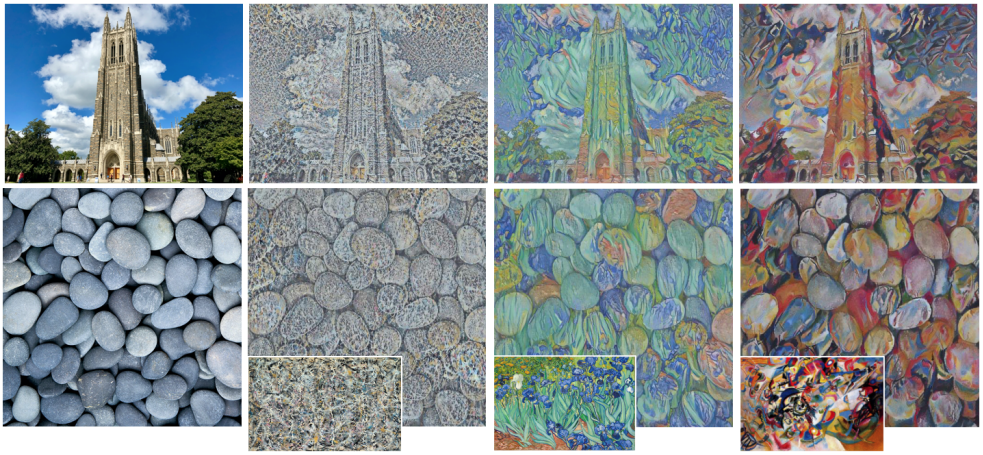

Figure 3: Comparison of content and style images of varying complexity. A photo of the Duke Chapel is shown on the top row while a photo of rocks is shown on the bottom row. Original content images are shown in the left column. Style images used to make the hybrid images are shown at the very bottom of the bottom row. Complex style images are able to transfer style but appear to be less aesthetically pleasing. The style images are, from left to right, Jackson Pollock's *Number One*, Van Gogh's *Irises*, and Kandinsky's *Composition VII*.

Shown in Figure 3, all styles were successfully transferred. When using the rocks which was considered a simple image, the Pollock abstract art and Van Gogh's *Irises* did not break any boundaries. However if we look at the rock photo combined with the style image of Kandinsky's *Composition* VII we see that rocks bleed color into each other. This is likely due to the fact that the style images where bleeding was not as observed contained defined boundaries whereas Kandinsky's *Composition VII* appears to have much less distinct boundaries and the photo does not really contain any objects. Pollock's work also does not contain objects but is filled with lines and best keeps the rocks shapes without combining any of the rocks. Furthermore, edges are the least sharp in the Duke Chapel photo combined with *Composition VII* than in the other two hybrid images. Edges are sharp in some places when the Duke Chapel was combined with *Irises* and sharp all over when combined with *Number One*. Objects with consistent patterns such as *Irises* appear to transfer style better than artwork more abstract pieces such as *Composition VII*.

3.3 Different Initialization Images

To test the effect of different initialization images, content, random noise, and style images were used. All hyperparameters settings previously mentioned were used except for the initialization image.

When initializing with the content image, as seen in Figure 4, the content image maintained almost all of its shapes and global structures. When initializing with the style image, there was a vast increase in the amount of overall style in the image. Along with the increase in style, objects from the style image also appeared. In the Shipwreck styled image, waves are clearly seen in the hybrid photo (Fig. 4, top right). In the hybrid image containing *Starry Night* and Tuebingen Neckarfront, we can see that the black mountain like object from *Starry Night* was fully transferred. The content structures are still apparent in the images initialized with style although they appear to be more in the background of the photos. For images initialized with random noise, there appears to be a great mixture of both content and style, both appearing in different segments in the image. The photos in general are a lot less comprehensible unless looked at closely. It is possible that more iterations may have lead to a clearer image.

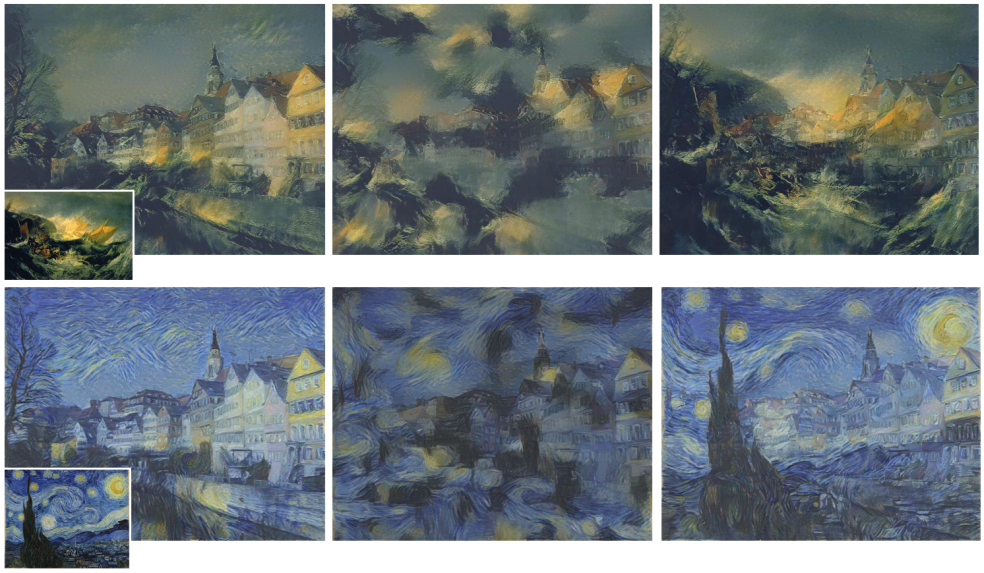

Figure 4: Initializing with different starting images. Content initialization is shown on the left column, random initialization is shown in the middle column, and style initialization is shown in the right column.

4 Conclusions

Accurate Neural Style Transfer was achieved as styles were able to be transferred to content images. In general, my reproductions of Gatys *et al.*'s work did not have as much style but this is something I can likely tune by upping my β hyperparameter. Proper style transfer was achieved over different content and style image complexities. There is no real way to define how well style transfer worked other than whether an output image is aesthetically pleasing, which can be quite subjective depending on what a user wants out of his or her neural style transfer. Different initialization images played a large role in defining the appearance of an outcome image as whatever the initialization image was contributed the most representation to the overall image. Random noise left a less comprehensible image although running more iterations or using a simpler style photo may be able to perform better. The images used for style when testing initializations had a great amount of visible brushstrokes and overall "texture" which may have been what caused the mess when using random initialization, indicating that random initializations are more prone to creating messy images. Overall, it was shown that content and style of a given image in a convolutional network are indeed able to be separated and can be combined to produce hybrid works of art.

In the future, I hope to continue working on this project to produce clearer photos by adjusting style weights, adjusting the α and β hyperparameters, and running more iterations to reduce the loss further. I also hope to be able to reduce the overall run time of the algorithm and eventually make it work in real-time.

References

[1] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. SIGGRAPH '01, page 341–346, New York, NY, USA, 2001. Association for Computing Machinery.

[2] Alexei Efros and Thomas Leung. Texture synthesis by non-parametric sampling. In *In International Conference on Computer Vision*, pages 1033–1038, 1999.

[3] Leon A. Gatys, Alexander S. Ecker, and M. Bethge. A neural algorithm of artistic style. *ArXiv*, abs/1508.06576, 2015.

[4] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

A Timeline and task allocation

Week 1

Read more papers focusing on Neural Style Transfer and learned what the newest research being done was. Established style and transfer images that I wanted to use. Got the pre-trained VGG edited to match the model in the paper.

Week 2

Read more about the Gramian matrix and implemented the loss functions along with the style weights and α and β hyperparameters.

Week 3

Finished up model and tested the effect of different initializations. Figured out a normalization issue I was having. Did various checks for success and implemented my own images to combine.

Week 4

Finished up making figures and created poster and wrote paper.