Artistic Image Transformation using CNN

Changhyun Lee (cl4017), Yunjeon Lee (yl7143)

May 9, 2023

Abstract

Artistic image style transfer is a technique that has gained popularity in recent years. This project focuses on exploring the use of Convolutional Neural Networks (CNNs) to extract high-level content features from an image and transfer the style of a source image to a target image while preserving the target's content features. Traditional methods utilize low-level features like image intensity to perform texture transfer, but we aim to improve this by using CNNs to generalize the equivalent style over multiple target images. By applying our proposed method, we aim to show that it can produce visually pleasing and convincing style transfer results. Our work contributes to the development of computer vision techniques for artistic style transfer and its practical applications.

Introduction

The importance of this research exists in exploring the information derived from the neural network model's training and structure. The ability to extract specific image characteristics such as style or content can lead to many applications in other fields that require deeper level of information from the visualization. To demonstrate the effectiveness of our approach, we use well-known artworks as source images and transfer their styles to target photograph images, allowing for easy perception of the resulting transformations. This project aims to implement the study by Gatys, Ecker, and Bethge to gain a deeper understanding of the mechanisms involved in extracting image features and transferring them to a new style.

Method/Approach

The implementation requires two types of images as input: the original image and the style image that represents the desired style. Using these images, the model generates an output that combines the content of the original image with the style of the style image.

To achieve this, we utilize Convolutional Neural Networks (pre-trained VGG16) to extract high-level information from the images. The model takes the original and style images as inputs and produces a new transformed image as an output. Our implementation focuses on replicating the model proposed by Gatys et al. using TensorFlow and Keras.

More specifically, we load the stacked input, style, and transformed data onto the VGG16 model and extract the layers for optimization. Content loss calculates the distance between the input and output images by analyzing "block2_conv2" layer. Style loss calculates the distance between style and output images using multiple chosen layers from VGG16. Total variation loss is a regularization to denoise the output image. These three loss are optimized using SciPy's "fmin_l_bfgs_b" algorithm. We train this model on Nvidia T4 GPU.

The output images generated by the above model will be evaluated based on several criteria such as the visual quality, accuracy of style transfer, and the retention of style from the source image. We compare artistic style transfer with photorealistic style transfer by utilizing photographs as source images and finding the constraint of the source's stylistic impact on the target's content features. Our implementation will demonstrate the effectiveness of the CNN-based approach for artistic image style transfer and provide insights into the inner workings of the model.

Experiments

In our experiment, we selected two images, one as the input image and the other as the style image. In order to ensure accurate comparison, we reshaped and normalized both of these images allowing us to extract information on a consistent scale. We utilized VGG-16, a convolutional neural network renowned for its 16-layer architecture. The model was trained on these images using content loss, which measures the difference between the transformed image and the features of the original image. This content loss serves as a key metric to guide the model's training process enabling it to generate the transformed image.

In addition, we experimented with two methods for the model. The first method is to experiment with various images and the second method is to experiment the same image with various parameters for the CNN model. We will discuss the results in the next section.

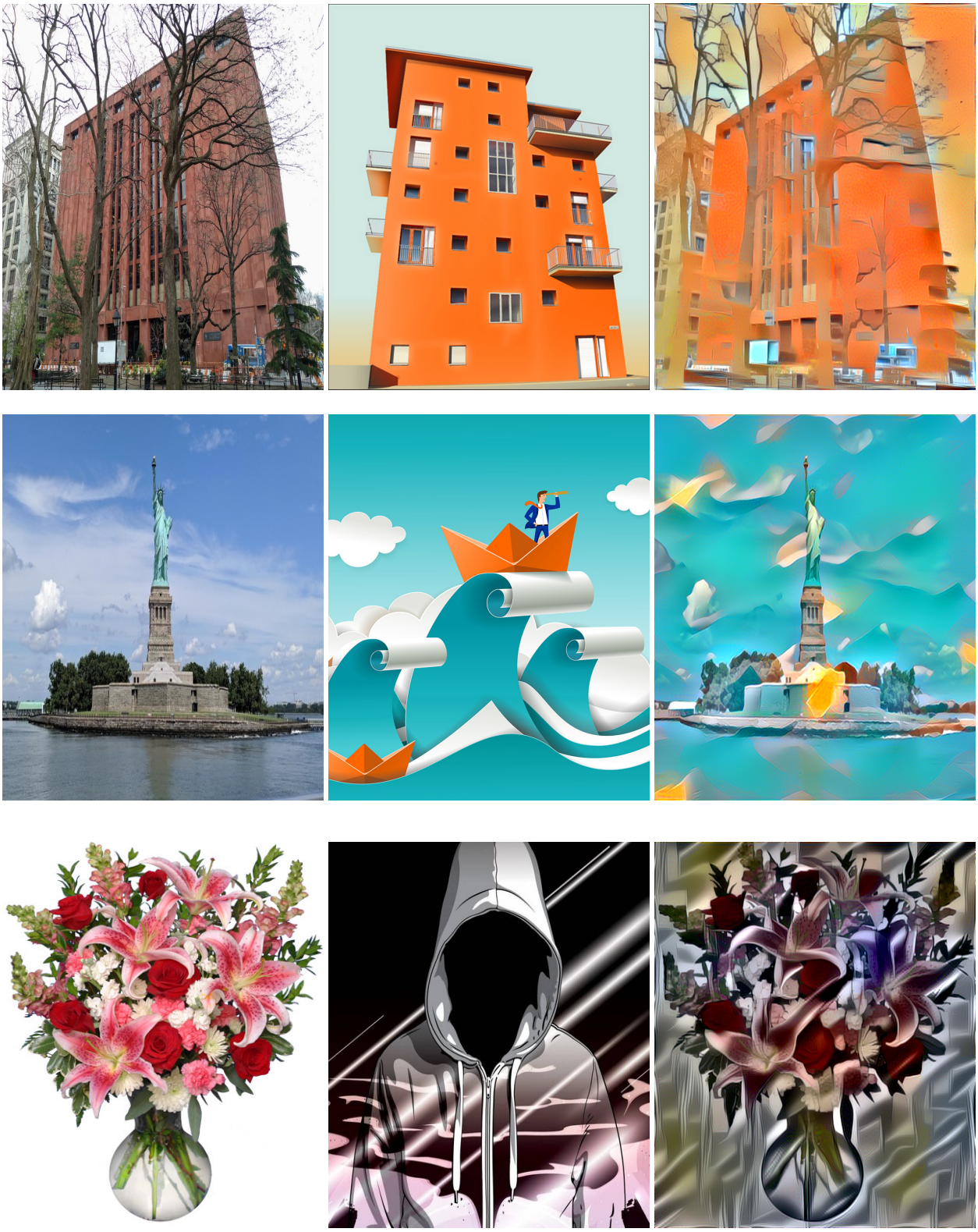
GitHub: `https://github.com/NYU-CDS-MS/Artistic-image-transformation-using-CNN`

Results and Conclusions

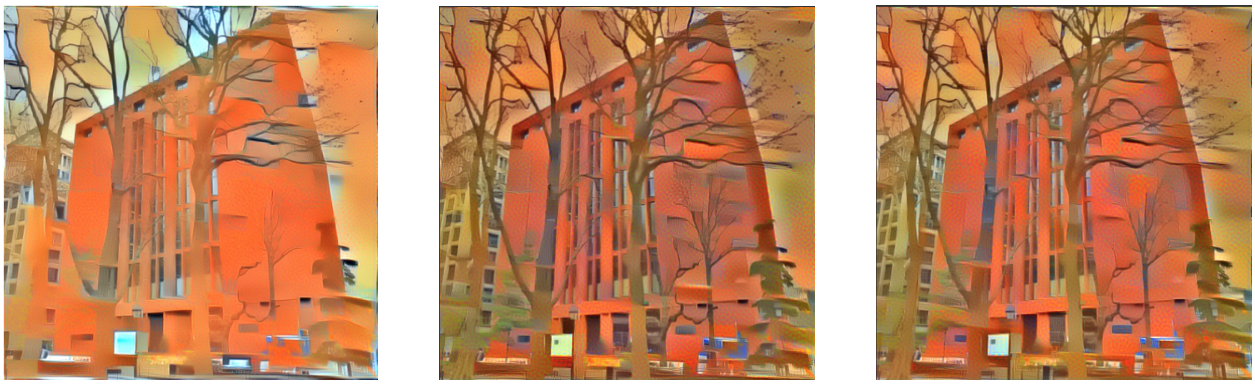
Our loss function successfully found the optimal model for our choice of style layers. For each group of style layers, the loss significantly decreased during the 10 iterations we ran, indicating the correct progress of optimization using three loss components (content loss, style loss, regularization).

```
Input image: Bobst_lib
Style image: vecbuildfinal
Style layes: ["block1_conv2", "block2_conv2", "block3_conv3", "block4_conv3", "block5_conv3"]
                    Iteration: 0 with loss: 137106595840.000000
                    Iteration: 1 with loss: 62104428544.000000
                    Iteration: 2 with loss: 51286859776.000000
                    Iteration: 3 with loss: 46661132288.000000
                    Iteration: 4 with loss: 43681615872.000000
                    Iteration: 5 with loss: 41488760832.000000
                    Iteration: 6 with loss: 39683702784.000000
                    Iteration: 7 with loss: 38031712256.000000
                    Iteration: 8 with loss: 36494548992.000000
                    Iteration: 9 with loss: 35134816256.000000
```

Below images are results from our experiments with various images. For each line, the first image is the original image and the second image is the style image we will use for training the model. Lastly, the third image is the transformed image using the first and the second image.



Choice of VGG16 style layers for loss	Image Number	Qualitative Style Effect
["block1_conv2", "block2_conv2", "block3_conv3", "block4_conv3", "block5_conv3"]	Image 1	These layers maintains the style image content the best while keeping styles from input image the least. Bobst building is not clearly separated from other buildings, and some of the objects from ground are difficult to distinguish. However, it applies styles from the style image with very similar colors and line characteristics.
["block1_conv1", "block2_conv1", "block3_conv2", "block4_conv2", "block5_conv2"]	Image 2	These layers maintains the input image content the best while keeping styles from style image the least. Bobst building is clearly distinguishable, with object from ground clearly visible. However, the colors from style image are not applied well, but line characteristics of style image exists.
["block1_conv1", "block2_conv1", "block3_conv1", "block4_conv1", "block5_conv1"]	Image 3	These layers moderately mixes together the input image content and styles from style image. Bobst building maintains its shape and separation from other buildings. Also, it applies styles from the style image moderately. Color seems to be the mix of input and style image.



(a) Image 1 (b) Image 2 (c) Image 3

Figure 1: Images with various layers

Overall, we believe our model transfers the styles of the one image to the other with accuracy, easily recognizable through human eye.

Discussions and Related Work

The work we did contrasts with previous approaches to image processing in that it makes use of image representations derived from CNN which is optimized for object recognition. These representations are able to explicitly represent high-level information in images and allows the model to separate and recombine image content and style. This approach is meaningful in that this improves the difficulty of separating image content from the style.

Challenges

We encountered challenges while constructing and training VGG16 model with the images. First, there are 16 different layers for VGG16 that we had to analyze and compare. There were no clear guidance on which layers represent style, content, and etc. However, we based our model on recent CNN research papers to finalize on the model layers to optimize for style, content, and regularization. Secondly, this model took over 10 minutes to train per iteration using the local CPU. Therefore, we had to incorporate utilizing GPU in order to reduce the runtime for faster results. With the Nvidia T4 GPU, our 10 iteration of training took 3 minutes to complete, which was a significant runtime improvement.

Future Scope

We believe utilizing more complex CNN architectures will likely to result in better retention of the styles. Since more complex models will contain larger, more accurate information about the source image, style and content features will also have improvements in their qualities. On the other hand, these models will likely be able to better separate the features such as styles, contents, etc. This can greatly affect the ability to transfer the amount of internal information while preserving the contents of the style-tranferred image.

References

[1] Gatys, L. A., Ecker, A. S., Bethge, M. (CVPR 2016): Image style transfer using convolutional neural networks. Available at: `https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf`