VGG19 TRANSFER LEARNING: IMAGE STYLE TRANSFER

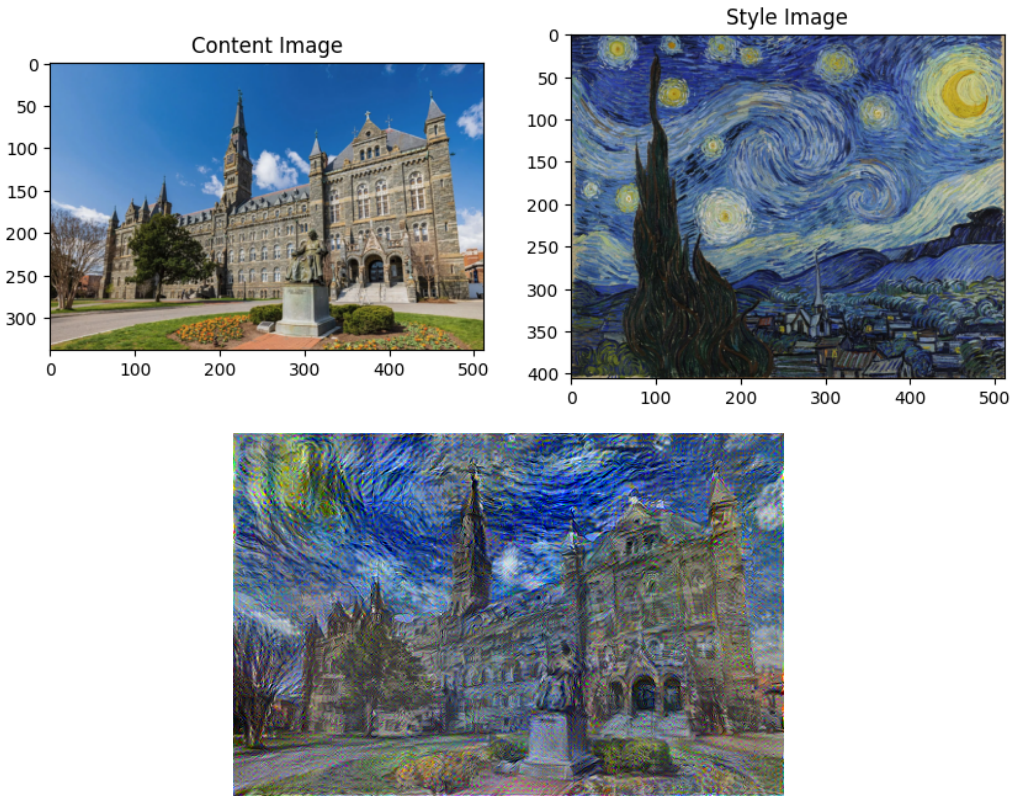
Mingqian Liu, Xin Xiang, Tianqi Zhou, Yanfeng Zhang

Georgetown Univerisity

December 2023

ABSTRACT

This report presents a study on image style transfer using the pre-trained VGG19 network, focusing on applying the aesthetic styles of celebrated artworks to photographic images. The VGG19's layers were selectively employed and fine-tuned with optimal content and style weight parameters to achieve the desired synthesis. Van Gogh's "Starry Night" and Georgetown University's Healy Hall were the primary images used for demonstrating the technique. Extensive hyperparameter tuning and a training regimen of 1000 epochs yielded a stylized image with markedly reduced loss, showcasing the model's ability to effectively blend artistic styles onto real-world structures while preserving the integrity of the content image. The result is a synthesized image that exhibits a harmonious fusion of art and reality, illustrating the potential of CNNs in creative image transformation.



Keywords Deep Learning · Transfer Learning · Image Style Transfer · VGG19

1 Introduction

This report presents a project that applies transfer learning to achieve image style transfer using the pre-trained VGG network. Transfer learning allows the utilization of a pre-existing neural network model, which has been trained on a large dataset, to perform new tasks. This method is efficient as it bypasses the need for training a network from scratch, leveraging the learned features for related tasks—in this case, image style transfer.

1.1 VGG Network

VGGnet, named after the Visual Geometry Group at the University of Oxford, is a deep convolutional neural network (CNN) known for its architectural simplicity and robust feature extraction capabilities. Designed by Karen Simonyan and Andrew Zisserman. The depth of VGGnet, with variants such as VGG-16 and VGG-19, refers to its 16 and 19 layer configurations, combining convolutional and fully connected layers—for example, VGG-16 is composed of 13 convolutional layers followed by 3 fully connected layers. The architecture standardizes on small 3x3 convolutional filters throughout, allowing it to capture complex features at various levels while maintaining a manageable number of hyperparameters. This consistency in layer design renders VGGnet particularly well-suited for transfer learning, where features from initial layers—trained on large, diverse datasets—can be repurposed for new tasks with minimal adaptation.

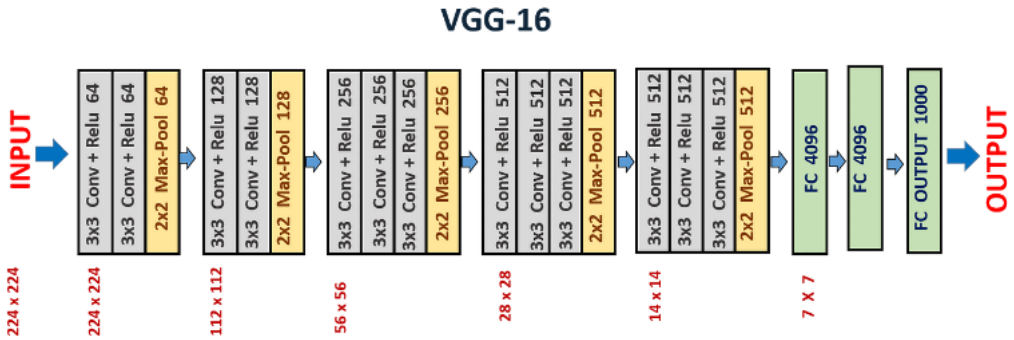


Figure 1: VGG16 Network Structure

1.2 Project Intention

Image style transfer is the process where the stylistic elements of a particular artwork, referred to as the style image, are imposed onto another image, known as the content image. The outcome is a synthesized image that retains the original content but displays the artistic

style of the reference artwork. For the purposes of this report, Georgetown University's Healy Hall is used as the content image, while Van Gogh's iconic 'Starry Night' serves as the style reference. The expected result is a rendition of Healy Hall as if painted by Van Gogh, demonstrating the practical applications of combining deep learning with artistic creativity.

2 Preprocessing

The preprocessing stage involves preparing images for the VGG19 model and choosing the right layers within the model. This process is split into two main areas: image preprocessing and layer selection.

2.1 Image Preprocessing

The images need to be formatted correctly for use with the VGG19 model. This includes adjusting the image size, color, and format. Similarly, when receiving images from the model, the image must be reverted or denormalized for clarity and interpretability.

For input image preprocessing, the images undergo a standard process to ensure compatibility with the VGG19 model. This includes adjusting all images to 224x224 pixels, aligning with the model's input size requirements. The RGB color components of each image are then normalized to match the standards set by the ImageNet dataset, with mean RGB values of [123.68, 116.779, 103.939], ensuring color fidelity. Finally, the images are converted from RGB to BGR format, consistent with the original training configuration of the VGG19 model.

In the output image preprocessing stage, the images are denormalized to enhance their interpretability and visibility. This involves re-adding the average RGB value to each color channel, which was subtracted during input preprocessing, to restore the original colors. The image format is reverted from BGR to standard RGB, the universal format for image display. Lastly, the pixel values, which were normalized to a range between 0 and 1 for input processing, are rescaled back to the standard 0 to 255 range for proper display.

2.2 Layer Selection

The VGG19 model's layer selection is crucial for capturing style features at different levels. Five style layers: '$block1\_conv1$', '$block2\_conv1$', '$block3\_conv1$', '$block4\_conv1$', and '$block5\_conv1$' are selected based on their ability to capture a range from basic details to broader patterns. These layers combine micro-level details with macro-level abstract

patterns, providing a comprehensive style representation. The selection of these layers is guided by previous research and experimental results, ensuring a balance between variety and the avoidance of overfitting.

For content, the '$block5\_conv2$' layer is chosen for its capability to identify complex patterns that represent the content more effectively. This layer balances capturing high-level content features with computational efficiency. It is particularly useful in style transfer, retaining the essential elements of the content image while adopting the style characteristics.

3 Method

3.1 Loss Function

The loss function in image style transfer forms the core of the algorithm, combining content loss and style loss into a weighted sum. Content loss quantifies the discrepancies between the synthesized and content images, ensuring the retention of essential details from the original content image during the styling process. Style loss assesses the adherence of the synthesized image to the style of the artwork, focusing on replicating the texture and color patterns without being confined to specific shapes. The total loss function is defined as below:

$$L_{total}(i,j,k) = \alpha L_{content}(i,k) + \beta L_{style}(j,k) \tag{1}$$

where i,j,k = feature maps.

3.1.1 Content Loss

Content loss adopts a squared loss function, similar to that the loss function in linear regression, to measure content feature discrepancies between the synthesized and content images. This is done through the Euclidean distance calculation. The content image, say, Healy Hall at Georgetown University, and the synthesized image are both processed via selected layers of the VGG network, producing corresponding feature representations (feature maps). These are then juxtaposed, layer by layer, to compute the cumulative content loss. The content loss function is defined as:

$$L_{content} = \sum_{l} \sum_{i,j} (\alpha C_{i,j}^{l} - \alpha P_{i,j}^{l})^2 \tag{2}$$

where i,j = feature maps, C = content image, P = synthesized image.

3.1.2 Style Loss

Style loss serves to penalize the synthesized image if its style diverges from the target style image. It employs the root mean square error to evaluate the differences between the Gram matrices of the two images. These Gram matrices, calculated from the VGG network's feature maps, encapsulate the distribution of features across the image layers. Both the style and synthesized images undergo this process, allowing for a pixel-by-pixel comparison that defines the style loss. The style loss function is defined as:

$$L_{style} = \sum_{l} \sum_{i,j} (\beta G_{i,j}^{s,l} - \beta G_{i,j}^{p,l})^2 \tag{3}$$

where i,j = feature maps, G = Gram Matrix, s = style image, p = synthesized image.

3.1.3 Total Loss

The composite loss function is defined as $L_{total} = \alpha L_{content} + \beta L_{style}$, with α and β being hyper-parameters that dictate the balance between content and style. These weights control the relative importance of content and style in the final result. Typically, the style weight surpasses the content weight to prioritize the style transfer aspect within the synthesized image.

3.2 Hyper-parameter Tuning

Hyper-parameter tuning is crucial in manipulating the balance between content and style in the synthesized image. The hyper-parameters α and β respectively control the preservation of content features and the influence of artistic style. For this project, α was held constant at a value of 1000 to stabilize the content feature influence, allowing for a focused adjustment of the style weight β.

A set of β values [1, 0.1, 0.01, 0.001] was methodically tested to determine their effect on the style transfer outcome. Each β value was subjected to 100 training epochs, and the resulting content loss, style loss, and total loss were recorded for analysis.

The selection of the optimal style weight was based on achieving the lowest possible losses. A style weight of 0.001 yielded the most balanced results, with the content loss recorded at 557,698.75, style loss at 410,616.06, and a combined total loss of 968,314.81. This tuning process is pivotal to fine-tuning the style transfer model, ensuring the final image demonstrates a harmonious blend of original content and applied style.

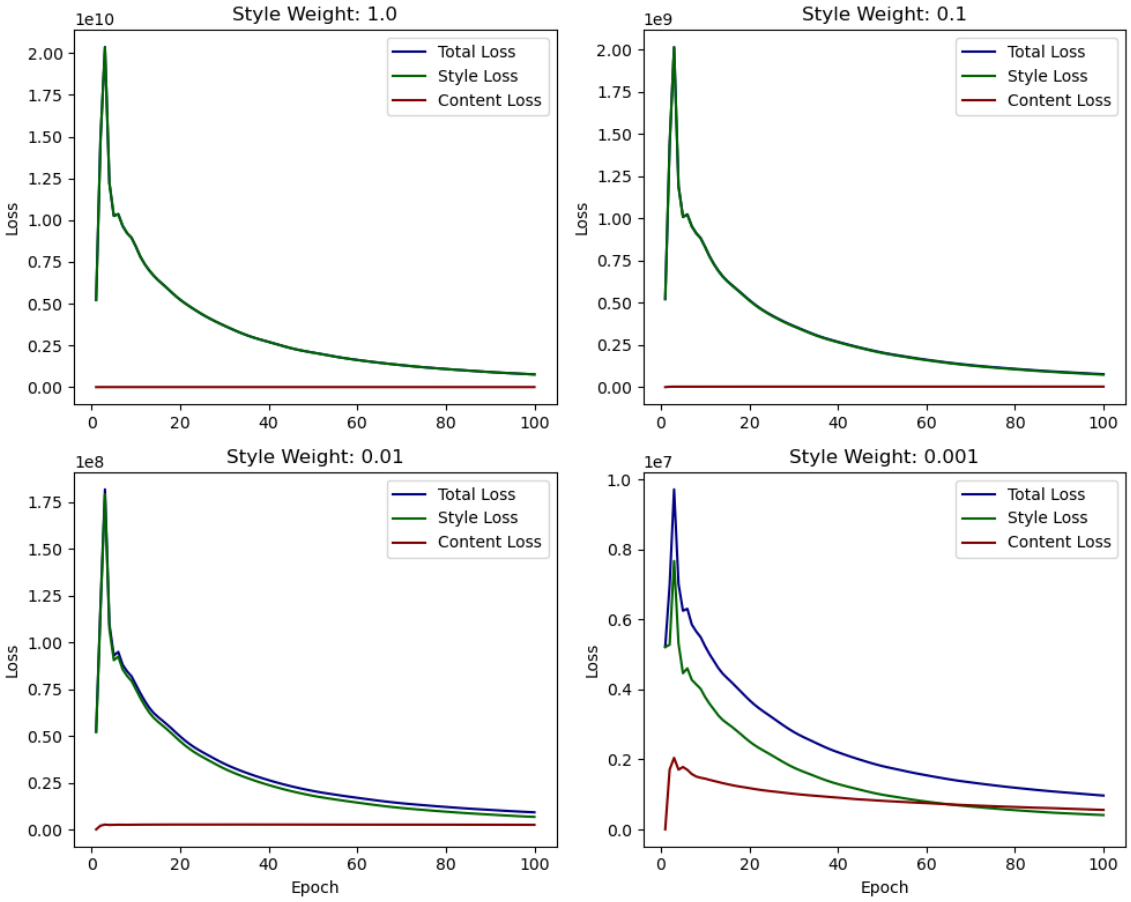


Figure 2: Loss trends during hyperparameter tuning of style weight [1, 0.1, 0.01, 0.001] over 100 epochs; the optimal performance is achieved with a style weight of 0.001, showing the lowest loss convergence.

3.3 Model Training

The training process for neural image style transfer involves iterative updating of the stylized image, similar to weight updates in a conventional neural network. The process begins with the initialization of the result image, which may involve introducing noise to the content image or using the content image itself as a starting point.

Following the principles outlined in the Loss Function section, content and style features are extracted from the synthesized image and compared with those from the original content and style images. This comparison facilitates the computation of losses.

During backpropagation, the VGG network's weights remain static, as the network is pretrained and not subject to further training. Instead, the pixel values of the result image

are adjusted based on the computed gradients of the loss. These updates are executed iteratively.

With each iteration, the synthesized image is refined to more closely emulate the style while retaining the content structure. This iterative process continues until convergence is reached, meaning that further iterations do not significantly reduce the loss.

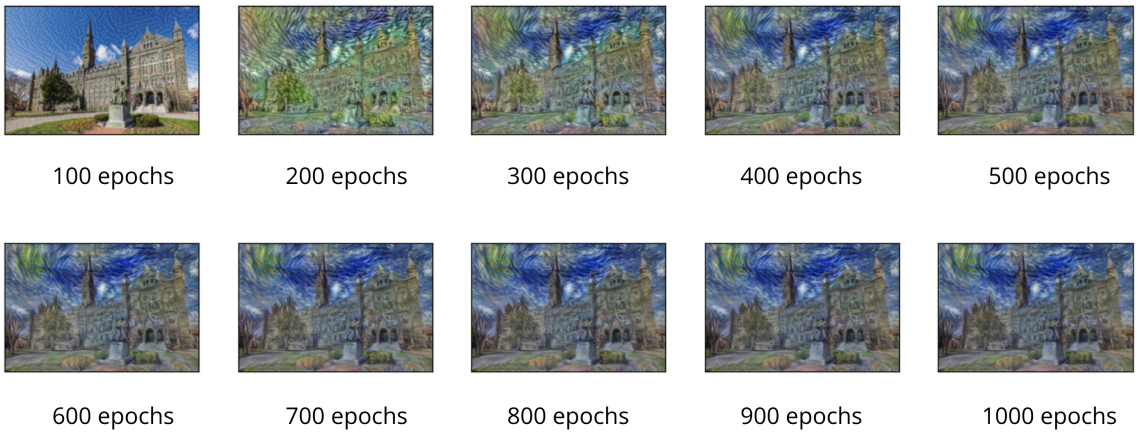


Figure 3: Evolution of the synthesized image across 1000 epochs, displayed at each 100-epoch milestone. Convergence is observed after 700 epochs.

4 Result

4.1 Optimal Model

The finalized model configuration, derived from the VGG19 architecture, employs a subset of its layers and is fine-tuned with a style weight of 0.001 and a content weight of 1000. This configuration emerged as the most effective following extensive hyperparameter tuning.

4.1.1 Model Performance

For the evaluative phase, Vincent van Gogh's "Starry Night" was chosen as the style reference due to its unique and widely acclaimed color scheme and brushwork. Georgetown University's Healy Hall was the content image, selected for its architectural similarities to the elements in "Starry Night", such as the sky, building, and foliage. This choice aimed to test the model's ability to apply Van Gogh's distinct style to real-world images effectively.

The performance metrics during the evaluation phase demonstrated the model's proficiency in style transfer. With 1000 epochs of training, a rapid decrease in total loss was observed, from 10,763,483 to 189,774.45, indicating quick convergence and the model's efficient

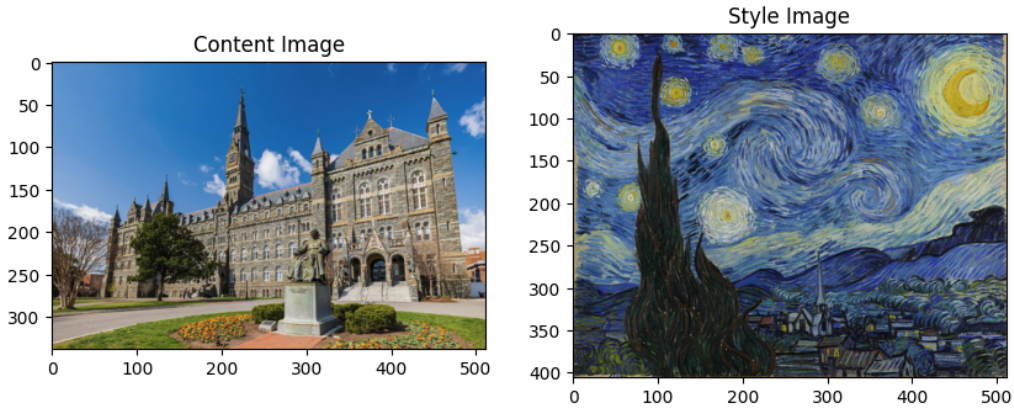


Figure 4: The content image and the style image for testing the model performance.

adaptation to Starry Night's artistic features. Both style loss and content loss decreased significantly from figure 4, confirming the model's capability to achieve a harmonious balance between mimicking the style of "Starry Night" and maintaining the content's authenticity, fulfilling the primary goals of the style transfer project.

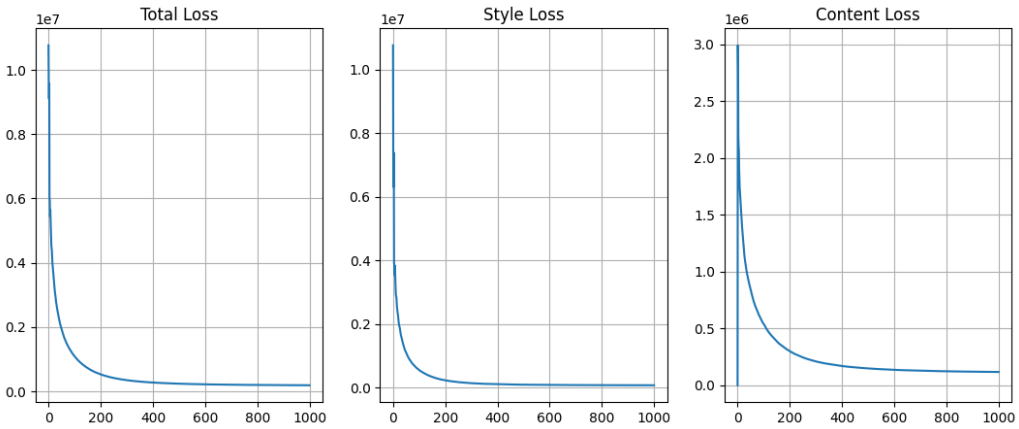


Figure 5: The line graphs show the loss metrics during the testing of the optimal model in the context of style transferring 'Starry Night' onto an image of Healy Hall.

Besides the notable decrease in loss metrics, the final output of the style transformation, transforming Healy Hall into a version inspired by "Starry Night", directly showcases the success of the model. This result is a clear demonstration of the model's ability to achieve the core aims of the style transfer project: effectively emulating the style of "Starry Night" while maintaining the authenticity of the original content.

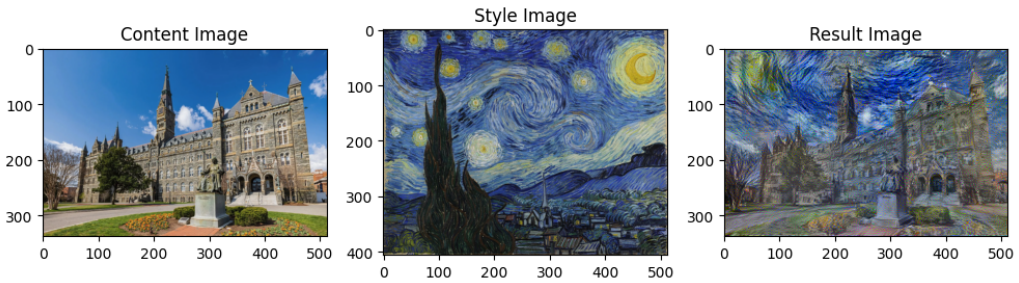


Figure 6: The final output of the style transformation.

4.2 Model Application

The output demonstrates the model's capability to transfer styles between two similar images, such as from 'Starry Night' to Healy Hall, which share common elements. To further assess the model's versatility, three diverse style images were applied to three distinct content images.

For a comprehensive test of the model's ability to capture varying artistic styles, selections included Vincent van Gogh's "Starry Night", representing colorful and distinct features; Roy Lichtenstein's "Bicentennial Print", exemplifying abstract pop art with bold shapes and vibrant colors; and Paul Signac's "Cassis, Cap Lombard, Opus 196", showcasing a gentle, dreamy impressionist style with soft brushstrokes.

Three different content images were chosen to evaluate the model's capacity to maintain unique content features. These included Healy Hall, representing architectural structures; the famous actor Robert Downey Jr. portraying Iron Man, capturing human expression; and Georgetown's official mascot, Jack the Bulldog, representing animal figures with a need for accurate depiction.

Figure 6 highlights the model's performance. Across all examples, the model not only captures the defining features of each artistic style but also ensures that the unique characteristics of the original content remain intact, from the architectural details of Healy Hall to the nuanced expressions of the human subject and the distinctive portrayal of the animal mascot. This demonstrates the model's adeptness in rendering artistic transformations while respecting the content's original identity, a testament to its robustness and sophistication in performing style transfers.

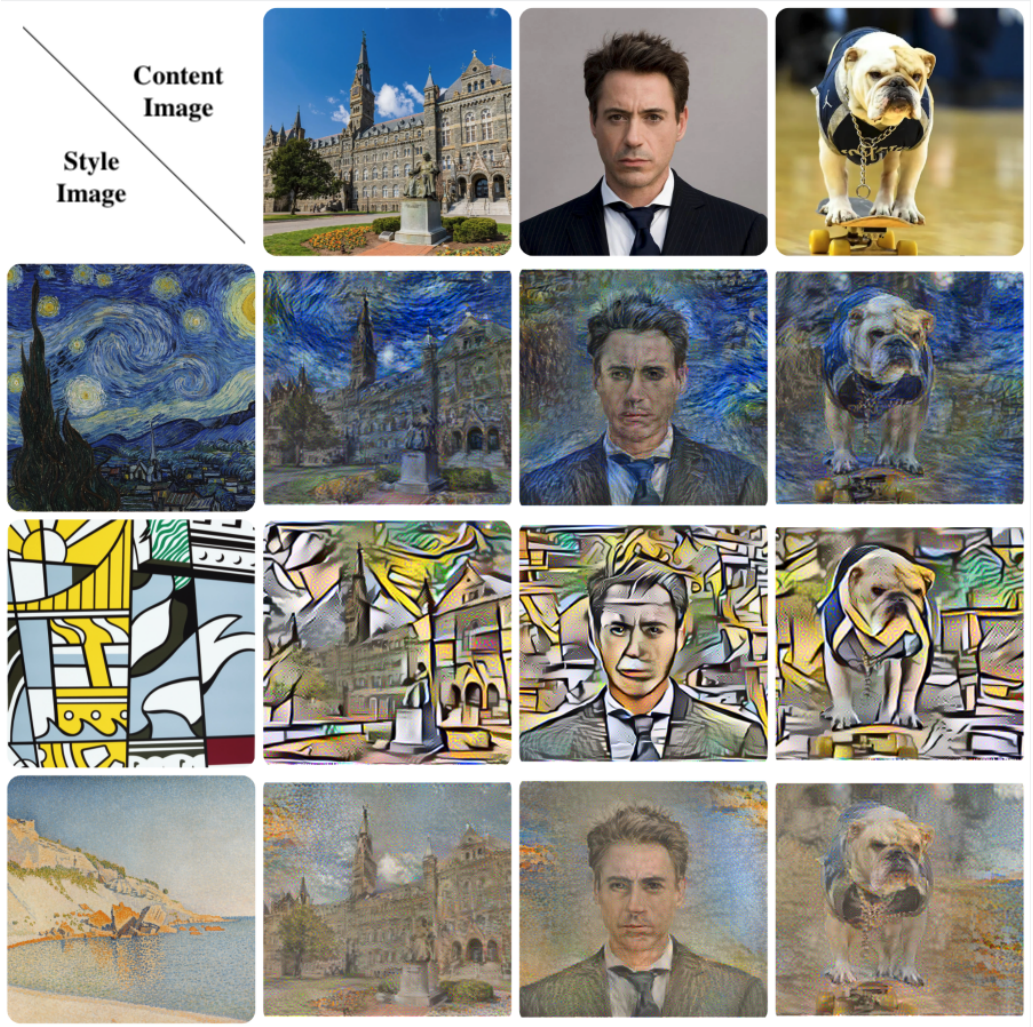


Figure 7: Multiple application of the style transformation model.

5 Conclusion

The multidimensional evaluation of the model reveals that a tailored configuration of the VGG19 architecture can effectively execute style transfers across a variety of themes and content. Adjusting the style weight has enhanced the model's ability to assimilate different artistic styles while preserving the essence of the original content. Although there are areas for improvement, particularly in capturing finer details as observed in the abstracted version of Healy Hall, these could potentially be addressed by optimizing the content weight. Overall, the high quality of the results affirms the model's proficiency, underscoring the success of utilizing VGG19 for transfer learning in producing comprehensive and high-quality style transformations.

References

[1] Babaev, Ruben, and Anna Plotkina. “A Quiz Game of Famous Painters.” Art Challenge, 2014, artchallenge.ru/?lang=en.

[2] Dumoulin, Vincent, Jonathon Shlens, and Manjunath Kudlur. "A learned representation for artistic style." arXiv preprint arXiv:1610.07629 (2016).

[3] Chollet, François. “Keras Documentation: Neural Style Transfer.” Keras, keras.io/examples/generative/neural_style_transfer/. Accessed 5 Dec. 2023.

[4] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[5] Jing, Yongcheng, et al. “Neural Style Transfer: A Review.” ArXiv:1705.04058 [Cs, Eess, Stat], 30 Oct. 2018, arxiv.org/abs/1705.04058.

[6] Singhal, Gaurav. “Gaurav Singhal.” Pluralsight, 3 June 2020, www.pluralsight.com/guides/implementing-artistic-neural-style-transfer-with-tensorflow-2.0.

[7] Zhang, Aston. “14.12. Neural Style Transfer¶ Colab [Pytorch] - Dive into Deep Learning 1.0.3 Documentation, Oct. 2020, d2l.ai/chapter-computer-vision/neural-style.html.