Texture Enhancement Using Neural Style Transfer

Mai Akram
Faculty of Engineering, Aisn Shams University
Cairo, Egypt
2401849@eng.esu.edu.eg

Mayar Ahmed
Faculty of Engineering, Aisn Shams University
Cairo, Egypt
2300305@eng.esu.edu.eg

Ahmed Fawzy
Faculty of Engineering, Aisn Shams University
Cairo, Egypt
2301486@eng.esu.edu.eg

Hazem Abbass
Faculty of Engineering, Aisn Shams University
Cairo, Egypt
hazem.abbas@eng.asu.edu.eg

Mahmoud Khalil
Faculty of Engineering, Aisn Shams University
Cairo, Egypt
mahmoud.khalil@eng.asu.edu.eg

***Abstract*— For instance, this work represents a method for generating artistic images with high perceptual quality using Deep Neural Network (DNN) based on the separable nature of image content and style within the Convolutional Neural Network (CNN). The adopted algorithm was based on Visual Geometry Group (VGG) Network, composed of a total of 19 layers. Meanwhile, the network was trained to minimize a loss function computed using the gradient descent, representing the trade-off between content representation and style representation. Indeed, the system was tested for applying the style of 15 well-known artworks to the content of 9 different images, with 3 captured results for each match, where the results were evaluated according to the structural similarity index measure (SSIM). Overall, the research proved to be efficient in the generation of the artwork images through the combination of the content from a photograph and the style of an artwork image with an overall high SSIM, high rate of total loss decay, and acceptable computational complexity. Hence, paving the way for more advanced applications within the computer vision field.**

Keywords*— *Neural Style Transfer, Deep Neural Networks, Convolutional Neural Network, Content Representation, Style Representation

I. INTRODUCTION

For instance, Computer vision systems have gone through vast developments after the introduction of the bio-inspired Deep Neural Networks (DNN), providing the capabilities to perform functions such as object and face detection and recognition. Still, the ability to create artistic pieces is known to be a unique capability of human beings. Meanwhile, since object recognition using DNN is based on the network hierarchy, where each layer is responsible for extracting specific feature maps of the image content, with higher layers resulting in the higher-level content in terms of the detected objects, their identity, and location within the image, described as the content representation, regardless of the individual pixel values. Likewise, style representation of input images can be extracted from the images using specially designed feature spaces for extracting texture information disregarding the global content [1]. In this process, known as style representation, the different layers extract the image appearance represented in its colors and local structure, with an increased complexity along the hierarchy. Accordingly, combining the two processes provides the ability to generate artistic images based on the content of one arbitrary image and the style of another well-known artistic work. Indeed, this paves the way for understanding the biological process of perceiving and creating artistic images, hence, facilitating the development of algorithms that operates in alike manner for interpreting and generating such images. This is based on the conclusion made by [1], that the content and style representations in Convolution Neural Network (CNN) are separable, allowing for their separation and recombination (Figure 1). Nevertheless, the content and style of an image are not completely separable since no single image can satisfy a perfect match of the demanded content and style. Hence, a weighing factor is assigned to each of the requirements, where, on the one hand, assigning more significant weight for the image content allows clear identification of its objects with a poor style matching. On the other hand, higher style weights generate a texturized variation of the artistic style, with weak content representation. Still, through tuning the weighing factors according to the output requirements, highly appealing image perceptual quality could be generated. Overall, the similarity between the generated image and the original one is calculated along with the total loss and the computational time as a performance metrics.

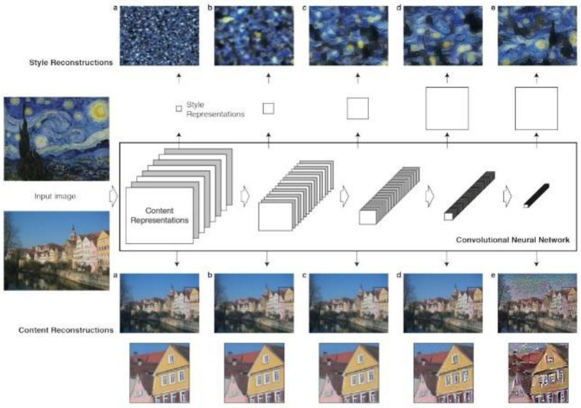


Figure 1 - The adopted VGG-Network structure as described by [1]

II. Literature Review

Previously executed work in image content and style separation was concerned with simpler subjects, such as handwritings and other small objects using Bilinear and Nonlinear models, [2][3],respectively. Other trials for artistic style transfer were based on non-parametric methods for directly altering the pixel representation according to the style [4][5][6]. For instance, the most common class of DNN used in computer vision is CNN, which consists of a group of layers with image filters, each extracting certain feature maps of the image, where the image is processed in a feed-forward method throughout the layer's hierarchy. One of the research projects concerned with image style that utilized DNN aimed to classify artistic works according to their time period through style recognition [7]. Likewise, [8] presented work on object recognition using CNN Visual Geometry Group (VGG) Network. Indeed, [1] studied the capability of algorithms in perception and representing art through utilizing DNN through combining content and image from two different sources using the VGG-Network, highlighting the potentials of such algorithms in a variety of computer vision areas, including psychophysics, functional imaging, and electrophysiological neural recordings.

III. Methods

A. VGG-Network

In this work, the CNN used was VGG-Network, composed of 19 layers, namely, 16 convolution layers and 5 average pooling layers for an enhanced gradient flow. On the whole, these layers consist of non-linear image filters with an increasing complexity as the image proceeds through the hierarchy. In addition, the number of filters increases as the image proceed through the hierarchy, while the image size decrease as a result of the down-sampling done by the max-pooling layers. For content reconstruction layers, higher layers the individual pixels are deprived while providing the overall image content as an output. Whereas for style reconstruction layers, the image style is perceived as a "stationary multi-scale representation" of the image through computing the correlation between the features at various CNN layers, until the artwork style is approximated by the generated image at higher levels of the hierarchy, though, with the cost of lost global image structure.

B. Loss Function

For instance, layer l with N_l filters, each with size M_l (filter width multiplied by its height), can have an overall response of $F^l \epsilon R^{N_l \times M_l}$. Accordingly, for comparing the generated image $\vec{x}$ information to that of the original image $\vec{p}$ in terms of the images content, a gradient descent is performed at each layer in terms of squared-error loss for filter i at position j in the layer l as follows:

$$L_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} \left(F^l_{ij} - P^l_{ij}\right)^2 \qquad \textbf{\textit{(1)}}$$

Where the gradient of the loss function is computed using standard error back-propagation as show:

$$\frac{\partial L_{content}}{\partial F^l_{ij}} = \begin{cases} \left(F^l_{ij} - P^l_{ij}\right) & if\ F^l_{ij} > 0 \\ 0 & if\ F^l_{ij} > 0 \end{cases} \qquad \textbf{\textit{(2)}}$$

Accordingly, the generated image is changed from a random one until a specified loss value is achieved. Specifically, the content reconstruction layers of the VGG are: $conv1\_1$, $conv2\_1$, $conv3\_1$, $conv4\_1$ and $conv5\_1$.

Whereas for the style matching, there exists a Gram matrix $G^l \in R^{N_l \times N_l}$ for layer l that consists of the correlations between various filters (i.e.: i and j) computed by the style representation on the top of the CNN, where the Gram matrix is computed as:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk} \qquad \textbf{\textit{(3)}}$$

Likewise, the gradient descent is computed for generating the image with the closest match to that of the demanded style through minimizing the mean-squared error between the Gram matrices of the artwork $\vec{a}$ and that of the new image $\vec{x}$ for all layers, each with a contribution to the total loss of:

$$E_l(\vec{p}, \vec{f}, l) = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G^l_{ij} - A^l_{ij}\right)^2 \qquad \textbf{\textit{(4)}}$$

Indeed, the total loss is calculated as:

$$L_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} \omega_l E_l \qquad \textbf{\textit{(5)}}$$

With ω_l representing the weighing factor of each layer, set to be $1/_n$, with n representing the number of active layers, which is 5 in this case. In a like manners, the gradient of E_l is computed using the standard error back-propagation as:

$$\frac{\partial E_l}{\partial F^l_{ij}} = \begin{cases} \frac{1}{N_l^2 M_l^2} \left(F^l_{ij}\right)^T \left(G^l_{ij} - A^l_{ij}\right) & if\ F^L_{ij} > 0 \\ 0 & if\ F^L_{ij} > 0 \end{cases} \qquad \textbf{\textit{(6)}}$$

Namely, the style reconstruction matching was performed between layers:

a) $conv1\_1$.
b) $conv1\_1$ and $conv2\_1$.
c) $conv1\_1$, $conv2\_1$, and $conv3\_1$.
d) $conv1\_1$, $conv2\_1$, $conv3\_1$, and $conv4\_1$.
e) $conv1\_1$, $conv2\_1$, $conv3\_1$, $conv4\_1$, and $conv5\_1$.

Overall, in order to combine the content and style from two different images, the total loss of the content and style are minimized jointly through their weighting factors α and β, respectively, according to:

$$L_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha L_{content}(\vec{p}, \vec{x}, l) + \beta L_{style}(\vec{a}, \vec{x}) \qquad \textbf{\textit{(7)}}$$

C. Content-Style Recombination

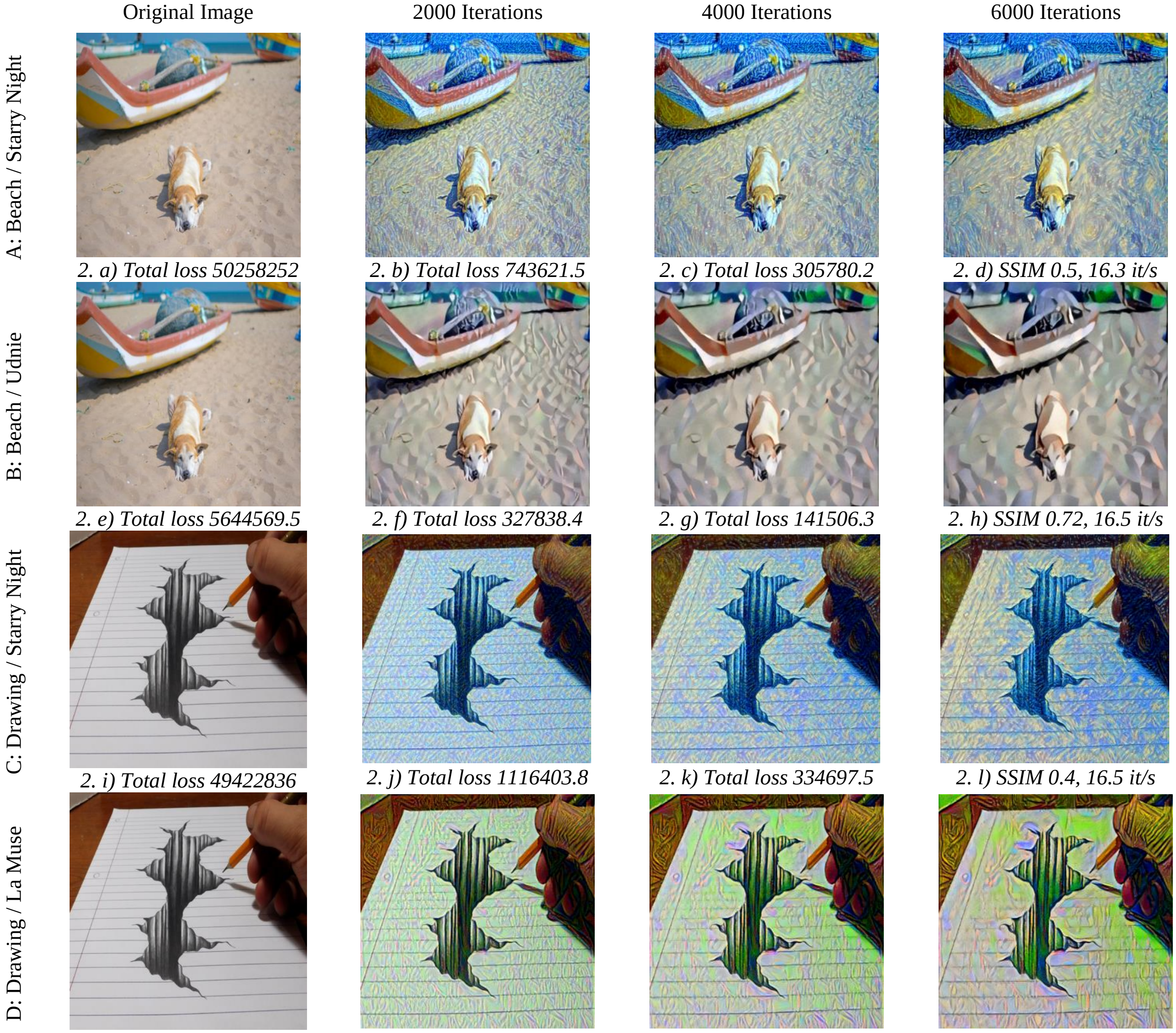
For this work, the weighting factors are given values of 1 and 0.01 for α and β, respectively. On the one hand, increasing the factor α was proven to result in a higher resemblance to the original image, while on the other hand, higher β generated a texturized image that closely captures the style of the artwork, though, with the trade-off of unrecognizable image objects.

D. Software Program

With regards to the software level, the program loops over nine arbitrary chosen images for a variety of scenes (e.g.: buildings, people, objects, natural views, etc.) applying the style of 15 well-known artworks to each of these images. The AI model iterated for 6000 iterations to generate each final output image. The generated image, combining the content of the original image and the style of the artwork, is captured during 3 states of conversion; after 2000 iterations (33% conversion), 4000 iterations (67% conversion), and 6000 iterations (100% conversion). For evaluating the results, the total loss is calculated for each generated image at its three stages, the average computational time is recorded in terms of iteration/s, and the Structural Similarity Index Measure (SSIM) of the final image with the original one is computed.

IV. Results

After applying the style of the 15 artworks to the 9 content images, the generated artistic image was captured at three different stages of the conversion (2 intermediate and 1 final image). Some of the resulting images are shown in Figure 2, including the total loss computed at each stage, the final achieved SSIM, and the average computational time. The figure shows different content-style combination at each row, whereas specific conversion stage is represented at each column.

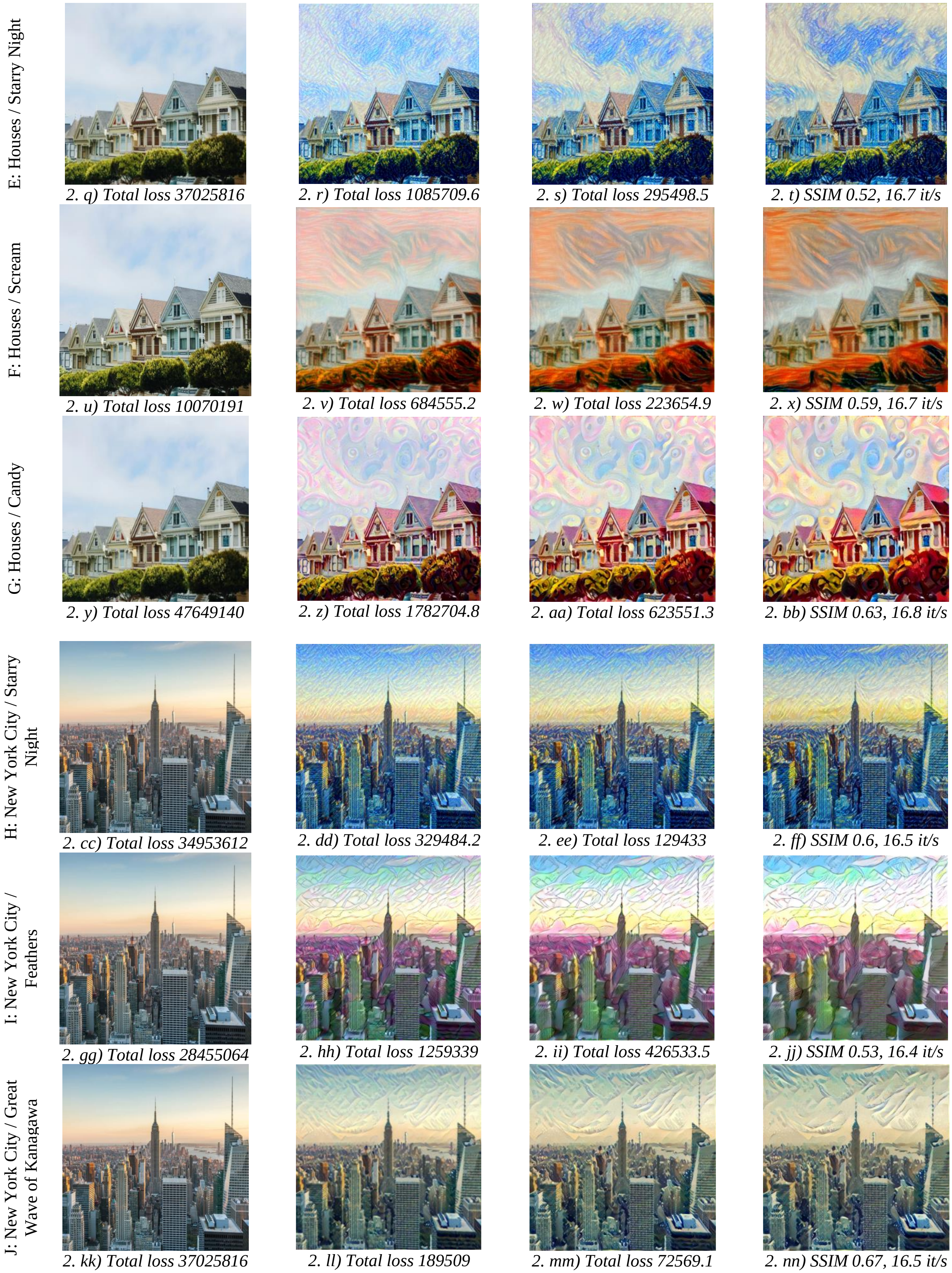


2. a) Total loss 50258252 *2. b) Total loss 743621.5* *2. c) Total loss 305780.2* *2. d) SSIM 0.5, 16.3 it/s*

2. e) Total loss 5644569.5 *2. f) Total loss 327838.4* *2. g) Total loss 141506.3* *2. h) SSIM 0.72, 16.5 it/s*

2. i) Total loss 49422836 *2. j) Total loss 1116403.8* *2. k) Total loss 334697.5* *2. l) SSIM 0.4, 16.5 it/s*

2. m) Total loss 62536888 *2. n) Total loss 1818776.1* *2. o) Total loss 666943.1* *2. p) SSIM 0.46, 16.5 it/s*


E: Houses / Starry Night
F: Houses / Scream
G: Houses / Candy
H: New York City / Starry Night
I: New York City / Feathers
J: New York City / Great Wave of Kanagawa

2. q) Total loss 37025816

2. r) Total loss 1085709.6

2. s) Total loss 295498.5

2. t) SSIM 0.52, 16.7 it/s

2. u) Total loss 10070191

2. v) Total loss 684555.2

2. w) Total loss 223654.9

2. x) SSIM 0.59, 16.7 it/s

2. y) Total loss 47649140

2. z) Total loss 1782704.8

2. aa) Total loss 623551.3

2. bb) SSIM 0.63, 16.8 it/s

2. cc) Total loss 34953612

2. dd) Total loss 329484.2

2. ee) Total loss 129433

2. ff) SSIM 0.6, 16.5 it/s

2. gg) Total loss 28455064

2. hh) Total loss 1259339

2. ii) Total loss 426533.5

2. jj) SSIM 0.53, 16.4 it/s

2. kk) Total loss 37025816

2. ll) Total loss 189509

2. mm) Total loss 72569.1

2. nn) SSIM 0.67, 16.5 it/s

Figure 2 - Artistic style transfer results over content of 4 photos and styles of 6 artworks

Overall, it could be perceived that various styles showed completely different effects on the same content image. Besides, some styles resulted in completely unrecognizable content at the final conversion stage, such as the effect of the style of the artwork “Feathers” on the image of New York City. Moreover, as more iterations proceeded, the image becomes a closer match to the style of the artwork, with a more continues visual experience and a smoother texture.

With regards to the image quality, the SSIM ranged between 0.4 (for the drawing image with the style of “Starry Night”) and 0.77 (for one of New York City images with the style of “Monet Umbrella” artwork). For instance, some styles proved to generally result in lower SSIM, such as “Mosaic”, whereas others generated images with high SSIM, such as “Monet Umbrella”. Likewise, some photographs, mainly photographs of faces or objects, had overall SSIM lower than others, such as those with buildings. In terms of the computational complexity, the recorded running time was mainly between 16.24 it/s (0.0616) and 16.84 it/s (0.0594 s/it) while using the GPU. Without the GPU, the average computational time was recorded as about 0.4 it/s (2.5 s/it). For the total loss, the images showed fast degrading of the loss function over the iterations.

V. Analysis and Discussion

Indeed, the research in consideration is concerned with developing a deep learning model that is capable of providing artistic images. Where the project proved to be able to:

1. Generate appealing artistic images through combining the content from an image and the style from an artwork.
2. Provide results with high quality in terms of the SSIM, up to 0.77.
3. Fast decaying of the loss function.
4. Operate in real time with GPU (up to 16.84 it/s), and relatively slow image processing without the GPU (average of 2.5 s/it).

Altogether, the model was capable of performing the separation process of the image content and style for the defined Weighting factors of 1 and 0.01 for content and style, respectively, with good computational time, acceptable loss function degradation, and high SSIM.

VI. Conclusion

Overall, the project results suggest that artificial neural systems are gaining the capability to perform complicated human-like functions, such as artwork creation, through content-style separation and recombination using CNN. This was achieved through training the model to optimize a loss function representing the trade-off between the similarity between the generated image and each of the content input image and style image. Besides, this task was successfully achieved with afforded computational complexity and with high output perceptual quality while minimizing the loss function. Moreover, this provides a method for deep understanding of the image perception and representation accomplished by neural systems, as well as the separable nature of content and style. Hence, the project successfully creates artistic images through being trained to separate the image content and style, just as complex biological vision systems, providing a step forward towards more advanced computer vision applications.

Future work involves:

1. Investigating the effect of utilizing different the network architecture (i.e.: Inception, ResNet, etc. instead of VGG-Network) on the generated results.
2. Optimizing the real-time operation of the algorithm.
3. Extending the application to 3D models and videos.
4. Testing the algorithm efficiency in real-life applications.

For further details and access to the code used in this research, please visit our GitHub repository at:

https://github.com/MaiAkram/Texture-Enhancement-Using-Neural-Style-Transfer

Whereas for accessing the presentation and video discussing the project, you can refer to the following drive:

https://drive.google.com/drive/folders/1DBXCKqoBRz-WDDDOznrFUMa0tXWSs6yA?usp=drive_link

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, “A Neural Algorithm of Artistic Style,” Sep. 2015, [Online]. Available: http://arxiv.org/abs/1508.06576

[2] J. B. Tenenbaum and W. T. Freeman MERL, “Separating Style and Content with Bilinear Models,” Hinton & Ghahramani, 1985.

[3] A. Elgammal and C.-S. Lee, “Separating Style and Content on a Nonlinear Manifold,” 2004.

[4] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, *Image Analogies*. ACM, 2001.

[5] M. Ashikhmin, “Fast Texture Transfer,” *IEEE Comput. Soc.*, 2003.

[6] A. A. Efros and W. T. Freeman, *Image Quilting for Texture Synthesis and Transfer*. ACM, 2001.

[7] S. Karayev *et al.*, “Recognizing Image Style,” Jul. 2014, doi: 10.5244/C.28.122.

[8] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Apr. 2015, [Online]. Available: http://arxiv.org/abs/1409.1556