Neural Style Transfer with Structure Preservation of Content Image

Anurag Singh Kotiyal
Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India

Nainika Choudhary
Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India

Dr. K.Senthil Kumar,
Assistant Professor (Sr. G)
Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India

***Abstract*— Neural style transfer is a class of algorithms that can manipulate digital images, to adopt the visual style of some other image. In simpler terms, we have two images – content image and style image. We take out the visual style from a style image and apply it to a content image, thereby rendering a new picture. Here style simply means, the brushstrokes, the texture, the patterns, etc. Up to the minute neural style transfer methods have not failed to provide astounding results by either training a convolutional neural network or using an optimization strategy that adopts an iterative approach. The original paper on style transfer used a VGG-19 model. Since the model used for style transfer was initially created for object detection, the feature that are extracted focus more on a central object, thereby remising other details. As a result, the synthesized image retains very slight structure of the content image. Due to this the generated image sometimes lacks the natural look of artist-made artwork. By means of this paper we would like to put forth a method to implement style transfer while retaining the structure from the content image. We hope to create an algorithm that can be used in a similar way to the famous Prisma filter on snapchat.**

Keywords*— *VGG-19, Neural Style Transfer, Convolutional Neural Networks, Deep learning, Transfer Learning etc.

I. INTRODUCTION

Since 1800’s painting gained a traction like it never did before. Famous artists like Van Gogh, Pablo Picasso and Rabindranath Tagore, to name a few, started emerging and spread their knowledge. This also led to the promulgation of a famous saying “A painting is worth a thousand words”. Deep Learning [1] has made a lot of things possible in the recent years. Even things that might have never been thought of getting done, in an automated way. An interesting advancement in the domain is known as Neural Style Transfer. We have two images and we synthesize a new image using them. This has made it possible for a non-artist to generate novel artworks that can rival masterpieces from paragons like Van Gogh or Picasso.

Deep Neural Networks [1], commonly abbreviated as DNN’s, have helped researchers in obtaining human-like efficiency on tasks performed by machines. They do so by imitating the decision-making mechanism of a human brain. It helps in providing solutions to a problem and also drawing conclusions from past experience, just like a human would do. A special class of networks known as Convolutional Neural Networks [2], [3] are best suited for tasks related to image processing. “A Convolutional Neural Network usually consists of multiple layers of minute computational units for processing visual information hierarchically in a feed-forward manner” [4]. The layers in these networks can be thought of as something that collects information at a certain level from an image. The sub-domain that deals with image related tasks is known as Computer Vision. In short, Computer Vision aims to automate tasks that a human visual system can accomplish [5], [6].

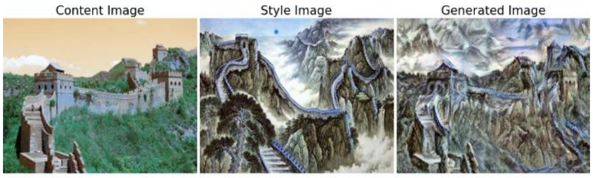

Fig. 1. Generating an artwork from the Great Chinese Wall (content image) and a piece of Chinese Artwork (style image) using neural style transfer

In Fig. 1, we can clearly see the style transfer algorithm in action. Using the Great Chinese Wall as content input and a piece of Chinese artwork as style input the system generates a novel artwork. Some common examples of image processing may include activities like denoising, and colorization. At times super resolution as well. Here the input is usually a degraded image which is noisy, or low-resolution, or grayscale, while a high quality colored images are obtained as the output [7]. Contrary to this, a computer vision task may include semantic segmentation or depth estimation of an image [7].

The original paper by Leon Gatys [4], [8] was the first paper that was published on the topic. It made use of a VGG-19 model and used feature representations from the first block of first, second, third, fourth and fifth layers respectively [4]. Obtained results were indeed good. Since then, a lot of improvements have been done on the algorithm and different approaches have yielded a variety of results. The most important aspect while working on a Neural Style Transfer algorithm is the loss function from style representation and content representation. A detailed study on these representation can be seen in the work of H.-H. Zhao et al.[9]. We propose to further study the latest trends that revolve around the problem and introduce an algorithm that can outperform some its predecessors.

II. STATE OF THE ART

A. *Literature Survey*

The following review will give the reader a clearer perspective on neural style transfer. In 2016, Gatys et al [4] proposed the first deep neural network based artificial system to create artistic images using a set of content and style images. This algorithm focuses on using convolutional neural networks and a feature space provided by a nineteen layered VGG-19 network - forming an output consisting of feature maps. They used a gram loss function and tried to minimize the loss generated from the newly created image, using the weighing factors of the content and style reconstruction. They used average pooling instead of max pooling to improve the gradient flow. Using the above methods, the author achieved appealing results.

Johnson et al [7] proposes a method to train a feed forward network for processing images using a perceptual loss function and perform a real-time image style transfer. They experimented on two tasks, image style transfer and super resolution in a single image, where they used perceptual loss functions instead of a per-pixel loss function as the latter depended on low level pixel information. However, the end results obtained were similar to the results obtained in “A neural algorithm of artistic style” proposed by Gatys et al [4] as measured by an objective function value. But the results generated in this paper were three times faster than the prior for image style transfer and the results for single image super resolution had visually appealing outputs for 4x and 8x super resolution.

Further Gatys et al [4], [8] proposed two methods to overcome the problem of preserving the colors of the original image found in the paper “A neural algorithm of artistic style” - color histogram matching and luminance-only transfer, while comparing the two methods and discussing the advantages and disadvantages of each. In the color histogram matching they compared two ways of transferring color of the content image with reasonable differences in the outputs, in the first way they transferred the color before style transfer and in the other they transferred the color after style transfer. They found the prior to give better results. In the luminance-only transfer method they used the YIQ color space to represent the color information of both the images, they also extracted the luminance channels from both the images and performed style transfer to form an output luminance image. Zhao et al [9] also proposed a new and different method to perform neural style transfer for better structure and color preservation and to prevent from creating artifacts using a content and style image. They have combined the local style losses with the global style losses, using a local loss approach to maintain the inherent styles of a style image and a global loss approach to maintain the structural information from a content image. Using this method, they have maintained the color and initial structure of a content image and reduced the making of artifacts.

Later researchers started to look for more applications of neural style transfer. Atarsaikhan et al [10] proposes a method to transfer style from one font to another by concentrating on the height, width and other factors and in result create new fonts. In addition, they demonstrate the effectiveness of character placement, orientation and weighted factors. They have used convolutional neural networks, a class of deep neural networks [1], to determine the structure of their content image and have used feature maps to determine the style of the style image. They have computed all their results using a 16 layered VGG-16 network. During their research, they faced a few challenges including styling single characters using multiple style images, deformations and distortions arising from noise.

Now the question arises on whether we can implement style transfer in real time for videos. E. Society [11] concentrates on applying the style of an image to a video in real time to achieve visually appealing results using a more time efficient optimization method. They have successfully maintained the style and temporal consistency of the frames using a feed-forward network. They calculate the temporal loss using a two-frame training method and combine the losses of the spatial domain and the temporal domain using a hybrid loss. They use pre computed optical flows to deal with the temporal loss obtained in the training stage and yield temporally consistent outputs.

Further, the paper on StyleBank [12] proposes to use a style bank which consists of multiple convolutional filter banks each consisting a different style. The flexibility achieved by the filter bank allows them to use an auto-encoder that renders an image in accordance to a particular style by producing an intermediate feature embedding rather than encoding the information from the styling image itself. This also helped them to fuse styles at both image and region levels at run time. Together with the help of an autoencoder and flexible representation of the filter bank they have stimulated further knowledge and understanding for image reconstruction and restoration.

The scope of style transfer is not limited to generating novel artworks only. Atarsaikhan et al [13] uses neural style transfer to create novel, unique, and genuine designer logos using clip art and text. They have concentrated on using style transfer only to a designated area to preserve structure of the texts and objects required to design a visually appealing logo. To preserve the shape and structure, a new loss function was proposed, which is based on a distance transform function. They have used this new loss function instead of cropping out the required image from the style image to create more natural results. Distance transform has helped in getting more dynamic and natural results as neural style transfer occurs within a confined region. They have stressed on the use of pre segmented content images to get better results in their experiment. They also observed that using a large weighing factor gives better results but a weighing factor too large can result in cropping of the style image.

There have been a lot of neural style transfer algorithms since 2016 but very few of them provide appealing results. Cheng et al [14] concentrates on retaining the basic structure and details of the original image while applying neural style transfer. They consider two factors to preserve the structure and details of their content image, a depth map and the image edges to represent the global and local structures respectively. The previous models were designed for object recognition and thus failed to recognize the details of an image and only concentrated on the central image, in this paper they have proposed to maintain the content structures and apply the style of the styling image while protecting the details of the content image. They notice that image representation holds a huge importance when it comes to styling and can generate images of different styles by using appropriate images. They also noticed that the global structure extraction network and local structure refinement network help in structure preservation and give better visual results as compared to previous proposed algorithms.

B. Inference from the Survey

This survey vividly portrays the advances in the domain of Neural Style Transfer. Not only has the technique be used on static images but also on real time dynamic videos. The technique has been developed for specific use cases too. We have seen logo generation [13], which is an amazing use case of the algorithm. We have also seen the creation of new font styles using style transfer [10]. All these papers clearly demonstrate the eclectic range of applications that can be achieved using style transfer. The paper on StyleBank [12] incorporated autoencoders to perform style transfer. Using perceptual loss function [7] Johnson achieved style transfer with similar results to Gatys [4] but with three times the speed of the latter. In fact, Jing et al [15] discusses extensively about all the applications of style transfer until 2019 and compares them in even more detail.

III. PROPOSED WORK

A. Description

Our proposed system will consist of a data pipeline for loading images, preprocessing them and passing them through the respective content layer and style layers of the VGG-19. The model will then calculate the style losses from the respective layers and run a summation over them to achieve a single style loss. Simultaneously, the content loss will be calculated using a content loss function, when the image is passed through the respective layer. We hope to develop a use case for generating human portraits using canvas paintings and images of individuals. Our model will be specifically fine-tuned for portrait generation. We also hope to retain the maximum possible structure from our content image so that the generated image doesn’t look disparate from the original. We will also try to transfer the colors and not just the styles from our style image to our newly generated image. We have seen so many applications of style transfer through our literature survey in Section II of this paper, but we did not see a single paper that focused on the specific use case of using style transfer for novel portrait generation.

The key note of our proposed system will be the natural aesthetic look, that our generated images will possess. We hope that our images will have a more natural touch to them as we will be focusing on retaining the original structure of content image while successfully accomplishing color transfer with our style transfer. As a result, the generated images should have the basic structure retained from a content image and the colors and style from a styling image. We have seen the effects of hypermeter tuning for style transfer in [9] and we hope to incorporate some techniques for our experimental research, so our model can be susceptible to lower quality style transfer for other pairs of content and style images like transferring style between the same city images, during different season. Further, we cannot guarantee that our model will work at par with other models for other settings of stylistic transfer, due to the selective nature of our research. We propose to use the VGG-19 model for our research.

We will be experimenting with some of the existing loss function for style transfer. In doing so, we would like to see the effect of different loss functions for portrait generation. A lot of the loss functions have been originally created to implement a special type of style transfer. For e.g., the distance transform loss function [13] was specially designed for the purpose of performing a contained style transfer by creating an image mask. Hence, we cannot use it for our research. We think that the original robust loss function from [4] should be a good starting point of our research. We hope to make the necessary changes at each step possible to improve the visual appeal of our generated results.

B. Merits

We hope that our model will be able to successfully transfer the style and color from our style image, while maintaining the basic structure from our content image. We have seen a lot of implementations of style transfer so far, but a lot of them lack the retention of original structure from the content image. Our main goal is to retain the initial structure from a content image while implementing style transfer. If we succeed in developing our model, we hope to provide an alternative that can be used just like the famous Prisma filter from snapchat.

IV. IMPLEMENTATION

A. Image Pre-processing and Deprocessing

This module deals with the basic pre-processing and de-processing of our content and style images. The first step is to load an image in the system and resize it to 512x512 pixels. Then we use the in-built VGG-19 pre-processing function to pre-process our images since we will be passing them through it. A small glimpse of what happens during the pre-processing via the function is that the values of 103.939, 116.779 and 123.68 are subtracted from the respective RGB color channels of an image and the pixel values become zero centered. This makes the entire calculation of losses from respective layers easier and faster. In the de-processing stage, we manually perform the opposite functions done in the pre-processing stage i.e., we add the values of 103.939, 116.779, 123.68 to the respective color channels. These values are the mean pixel values for the color channels in a VGG-19 model.

B. Content Loss

In this module we begin by calculating the loss between our content image and our generated image. In simpler words, we are just calculating the Euclidean distance, using the formula below, between the pixel values from our content image and the newly generated image. Then we do a summation over all the values to get a single output. Our goal is to reduce this value as much as we can and the smaller this value the closer our generated image will be to our content image. This means that the structure of our content image depends gravely on our content loss.

$$L^{l}_{content(p,x)} = \Sigma_{i,j} (F^{l}_{i,j}(x) - P^{l}_{i,j}(p))^{2}$$

x = input image

p = content image

$F^{l}_{i,j}(x)$ = feature representation of network with input x

$P^{l}_{i,j}(p)$ = feature representation of network with input p

C. Style Loss

In this module we will be dealing with the style loss between our style image and our generated image. We will be using a modified version of the gram matrix [4] for determining the loss between our styling image and the image that has been newly rendered. We will calculate the gram score of an image by multiplying its pixel values with their transpose. Then we need to calculate the variation between gram scores from our style image and our generated image. Our goal is to reduce the style loss to a lowest possible value. The lower the style loss, the more prominent the style transfer will be in our generated image.

$$E_l = \sum_{i,j} (G^l_{i,j} - A^l_{i,j})^2$$

$$L_{style}(a, x) = \sum w_l E_l$$

E_l = contribution of each style layer to style loss

N_l = number of feature maps

M_l = size of feature maps (height * width)

$G^l_{i,j}$ = style representation of input image x in layer l

$A^l_{i,j}$ = style representation of input image a in layer l

$w_l = 1/||L||$ = weighing factor

D. Total Loss

This is the last module of our model and it combines the style loss and the content loss from the model. It will do so, by calculating the sum of style weight times style loss and content loss times the content weight. It is important to note here that the style weight and the content weight are tunable hyperparameters. Depending on our application we can increase or decrease their values to obtain the desired results. A higher style weight will lead to more style in our generated image while a higher content weight will lead to more structure in our generated image as stated by Zhao et al [9].

$$style_{score} = (\sum^{s}_{i=0} w_i * L_{style}) * W_s$$

$$content_{score} = (\sum^{c}_{i=0} w_i * L_{content}) * W_c$$

$$total_{loss} = style_{score} + content_{score}$$

L_{style} = style loss

$L_{content}$ = content loss

w_i = weight per style layer and content layer respectively

W_s = style weight

W_c = content weight

E. Hyperparameters

We chose the second blocks of first, second, third, fourth and fifth convolutional layers from the VGG-19 for our style representation and the second block of the fifth convolutional layer for our content representation. Further, we used Adam optimizer for our research and the desired results were obtained by running the model for 1000 epochs. The style weight and the content weight differed by a factor of 1e5.

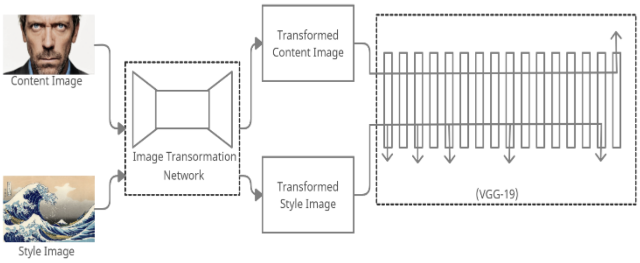

Fig. 2. Architecture Diagram

We can observe Fig. 2, for understanding the pipelined structure of the proposed model. It clearly depicts the role and positioning of various modules inside the proposed system architecture. First, we pick two images i.e., content and style. Then we preprocess them by passing through a transformation network. Next, we pass the transformed content and style images through the respective layers of a VGG-19. The new image is generated based on the losses obtained from the respective layers. This image is now optimized after every training step of the model.

V. RESULTS DISCUSSION

We were able to generate two types of results. Our first experiment achieved successful style transfer with preserved structure from a content image and perfectly transferred colors from our styling image. Contrary to this our second experiment yielded successful style transfer from style image accompanied by the color transfer. However, the results were not as visually appealing as our first experiment since the content structure was slightly distorted. The following figures will help in explaining our results better.

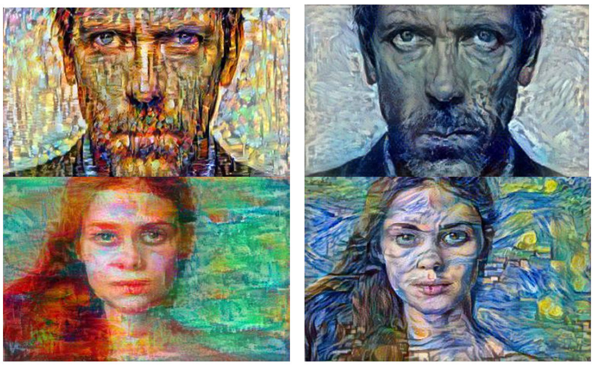

Fig. 3. Style Transfer with preserved structure

In Fig. 3, we can clearly see the results from our first experiment where we successfully implemented style transfer with color transfer from our style image and preserved the structure from our content image. The results were visually impressive, strongly maintaining the structure of our content image. In fact, style here is quite dominant. We picked up two random images from google that served as our content images and ran some experiments with various kinds of style images. In general, the style images with lighter colors resulted in a better style transfer.

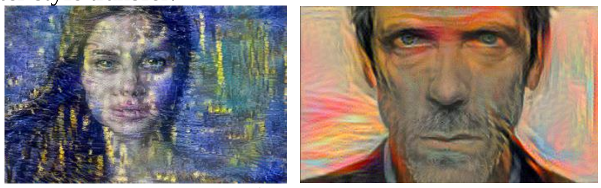

Fig. 4. Style Transfer with slight structure distortion

In Fig. 4, we can see the results of style transfer along with color transfer from our style image. Here the color has been successfully transferred from our style image to our generated image but the generated image does not retain the original content structure well. As a result, the generated image looks hazy. These results were seen more often when the style images had more solid colors and brushstrokes. Hence, from our experimentation it is evident that the results from style transfer with preserved structure were more aesthetic than the ones from a style transfer approach that distorted the content structure.

VI. CONCLUSION

We would like to conclude that our study was somewhat successful in creating natural portraits by means of synthesizing an artificial image using two images. However, there is still a lot of scope for improvement for further studies. In future, researchers can try to make the generated results smoother which will improve the natural touch of the generated image even further. There is a wide opportunity to experiment with different combinations of content and style images to generate better results.

REFERENCES

[1] Q. Wu, Y. Liu, Q. Li, S. Jin, and F. Li, "The application of deep learning in computer vision," *Proc. - 2017 Chinese Autom. Congr. CAC 2017*, vol. 2017-Janua, pp. 6522–6527, 2017, doi: 10.1109/CAC.2017.8243952.

[2] N. Jmour, S. Zayen, and A. Abdelkrim, "Convolutional neural networks for image classification," *2018 Int. Conf. Adv. Syst. Electr. Technol. IC_ASET 2018*, pp. 397–402, 2018, doi: 10.1109/ASET.2018.8379889.

[3] S. Albawi, T. A. M. Mohammed, and S. Alzawi, "Layers of a Convolutional Neural Network," *Ieee*, 2017.

[4] L. Gatys, A. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *J. Vis.*, vol. 16, no. 12, p. 326, Sep. 2016, doi: 10.1167/16.12.326.

[5] X. Li and Y. Shi, "Computer vision imaging based on artificial intelligence," *Proc. - 2018 Int. Conf. Virtual Real. Intell. Syst. ICVRIS 2018*, pp. 22–25, 2018, doi: 10.1109/ICVRIS.2018.00014.

[6] A. Rosenfeld, "Computer Vision: Basic Principles," *Proc. IEEE*, vol. 76, no. 8, pp. 863–868, 1988, doi: 10.1109/5.5961.

[7] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9906 LNCS, 2016, pp. 694–711.

[8] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman, "Preserving Color in Neural Artistic Style Transfer," pp. 1–8, Jun. 2016, [Online]. Available: http://arxiv.org/abs/1606.05897.

[9] H.-H. Zhao, P. L. Rosin, Y.-K. Lai, M.-G. Lin, and Q.-Y. Liu, "Image Neural Style Transfer With Global and Local Optimization Fusion," *IEEE Access*, vol. 7, pp. 85573–85580, 2019, doi: 10.1109/ACCESS.2019.2922554.

[10] G. Atarsaikhan, B. K. Iwana, A. Narusawa, K. Yanai, and S. Uchida, "Neural Font Style Transfer," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Nov. 2017, vol. 5, pp. 51–56, doi: 10.1109/ICDAR.2017.328.

[11] E. Society, "Real-Time Neural Style Transfer for Videos," vol. 50, no. 5, pp. 835–844, 2011.

[12] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "StyleBank: An explicit representation for neural image style transfer," *arXiv*, pp. 1897–1906, 2017.

[13] G. Atarsaikhan, B. K. Iwana, and S. Uchida, "Contained Neural Style Transfer for Decorated Logo Generation," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, Apr. 2018, pp. 317–322, doi: 10.1109/DAS.2018.78.

[14] M. M. Cheng, X. C. Liu, J. Wang, S. P. Lu, Y. K. Lai, and P. L. Rosin, "Structure-Preserving Neural Style Transfer," *IEEE Trans. Image Process.*, vol. 29, no. XX, pp. 909–920, 2020, doi: 10.1109/TIP.2019.2936746.

[15] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *arXiv*, pp. 1–20, 2017.