Pune Institute of Computer Technology
Dhankawadi, Pune

A SEMINAR REPORT
ON

Real-Time Neural Style Transfer

SUBMITTED BY

Isha Joshi
Roll No.:31338
Class:TE-3

Under the guidance of
Prof.D.D.Kadam

DEPARTMENT OF COMPUTER ENGINEERING
Academic Year 2019-20

DEPARTMENT OF COMPUTER ENGINEERING

Pune Institute of Computer Technology

Dhankawadi, Pune-43

CERTIFICATE

This is to certify that the Seminar report entitled

"Real-Time Neural Style Transfer"

Submitted by

Isha Joshi Roll No. 31338

has satisfactorily completed a seminar report under the guidance of Prof. D. D. Kadam towards the partial fulfillment of third year Computer Engineering Semester II, Academic Year 2019-20 of Savitribai Phule Pune University.

Prof. D. D. Kadam
Internal Guide

Prof. M.S.Takalikar
Head
Department of Computer Engineering

Place : Pune
Date : 17 April 2020

ACKNOWLEDGEMENT

I sincerely convey my gratitude to my guide Prof. D. D. Kadam, Department of Computer Engineering for her constant support, providing all the help, motivation and encouragement from beginning till end to make this seminar a grand success. She provided all the necessary help and gave the right direction towards completing this project.

I also sincerely thank our Seminar Coordinator Prof. B.D.Zope and Head of Department Prof. M.S.Takalikar for their support.

Contents

List of Tables

List of Figures

Abstract

The process of using Convolutional Neural Networks to render a content image in different styles is referred to as Neural Style Transfer (NST). It takes two images as input - a style image and a content image. The output generated is an image which has the same structure as the content image but it is rendered using the texture of the style image. The system uses feature representations in the intermediate layers of the network to separate and recombine content and style of the two images, using an algorithm for the creation of stylized images.

The stylized image is obtained by iterative optimization which is computationally inefficient[1]. This approach produces high quality images but it is slow. A variety of approaches are proposed to either improve or extend the original NST algorithm[1].

Using feed-forward generator networks[4], stylized images can be generated in a single pass. Along with this, the implementation of instance normalization[6] can significantly improve the quality of feed-forward style transfer models. The implementation presented in this research is based on a combination of the original model[1], perceptual losses in feed-forward network[4] and instance normalization[6].

Keywords

Convolutional Neural Network, Texture synthesis, Style Transfer, Deep Learning, Photorealism, Image stylization.

1 INTRODUCTION

The technique of generating an output image from two images- a content image and a style image is called as style transfer. The new image has the structural elements of the content image and the texture of the style image. It is the technique of recomposing images in the style of other images. It is technique of image stylization, which studied within non-photorealistic rendering.

The seminal work of Gatys et. al.[1] proposed the approach of Neural Style Transfer(NST) which made use of deep neural networks to generate the new image. Neural networks use different feature representations to display content and style of an image. This approach demonstrated the power of convolutional neural networks to process images in order to transfer artistic style of paintings to other images.

Convolutional Neural Networks (CNNs) are a type of Neural Networks that have proven very successful in image processing problems because of improved state of hardware combined with increased amount of data available. Pre-trained neural networks on a particular image dataset are capable of producing features for content and style of an image which gradually change over the layers of the network.

The original algorithm[1] had treated the task as an optimization problem which led to an inefficient solution requiring thousands of iterations to generate a single output image. To tackle this inefficiency, fast style transfer uses feed-forward neural networks[6]. Fast style transfer also uses CNNs but they train a model for a single style reference image which can generate the output images in a few seconds. Although a particular model is trained for only a single output image, it generates output in a single pass with a feed-forward network.

1.1 Motivation

Due to the increase in the computational power of hardware devices and with new technologies like style transfer anyone can create and share an artistic masterpiece[13]. Everyone is not born with artistic capabilities and style transfer gives a chance to discover the artist within a person. Artists can share their creative abilities to the world and people can use it to recreate these masterpieces. People all around the world can experiment with their own creativity using style transfer.

The study of neural style transfer could help understand how humans perceive images in an algorithmic way. Since neural networks are a representation of the human brain at work, studying style transfer can help us understand the way humans analyze images. It will also help us understand the working of CNN's better and explore its inner layers. It contains different feature representations for style and content of an image.

1.2 Applications

Neural style transfer can be applied in a number of ways:

- Photo editors
  Photo and video editing tools have an added edge with the ability of rendering regular photographs in the styles of famous paintings. It adds the touch of revered artists in ordinary photos and videos. With the increasing computation power, AI can now be embedded in mobile phones.

- Commercial art
  Through style transfer, AI can become more influential in people's daily lives. The production of high-quality paintings and unique advertising campaigns are just the few ways in which style transfer can influence commercial art.

- Gaming
  Stadia is a cloud-powered video game streaming service. This video game had the striking feature which allowed the gamer to render the virtual reality in different colour patterns and palettes. Such efforts can be monumental in helping the gamer discover the artist in themselves. Thus with the help of style transfer, those with no artistic capabilities can also experiment with art.

- Virtual Reality
  Virtual Reality is an emerging field with lots of exciting possibilities. There has not been much work in developing applications in this field. Style transfer can be of assistance to developers to tell visual stories in different ways through games and films.

2 LITERATURE SURVEY

The Following table shows the literature survey by comparing techniques propose in various references:

Table 1: Literature survey

Methods	Arbitrary style	Efficient	Learning free	Loss	Image Quality
Gatys et. al.[1]	Yes	No	Yes	Gram Loss	Good and usually regarded as best quality.
Johnson et. al.[4]	No	Yes	No	Perceptual Loss	Results are close to [1] but generated within real-time
Ulyanov et. al.[6]	No	Yes	No	Perceptual Loss	Results are close to [1] along with improved instance normalization
Chen et. al.[9]	No	Yes	No	Disparity Loss	Results are good but model size increases

3 A SURVEY ON PAPERS

3.1 Gatys et. al.[1]

Gatys et. al. observe that a pretrained convolutional neural network is able to identify the content and style parts of an image separately. Intermediate representations of the content and style images are derived from the immediate layers of the VGG-19 network. The structure of the new image is maintained by reducing the difference between the representations of the content image and the output image and the style is developed by comparing Gram matrices of the style image and the output image.

However, the algorithm of Gatys et al. does not perform well in preserving the fine structures and edges during stylization since CNNs lose some low-level information.

3.2 Johnson et. al.[4]

The approach suggested is to train a feed-forward network for a specific style image which can be used to develop output images in a single pass within a matter of seconds. The network used is similar to that of DCGAN[5] but with residual blocks and fractionally strided convolutions. The algorithm of Johnson et al. facilitates real-time style transfer but their design is basically similar to Gatys et. al.[1] and hence it suffers the same problems as well.

3.3 Ulyanov et. al.[6]

Ulyanov et. al.[6] and Johnson et. al.[4] share a similar idea, which is to pre-train a feed-forward network for a single style image and produce a stylised result with a single forward pass. Ulyanov et al. [6] further find that applying normalisation to every single image rather than a batch of images leads to a significant improvement in the accuracy of the output images. This single image normalisation is called instance normalisation, which is equivalent to batch normalisation when the batch size is set to 1. The style transfer network with IN achieves visually better results.

3.4 Chen et. al.[9]

In this method the idea is to use different network structures to analyse the style and content components of the image. The middle layer of the CNNs (StyleBank Layer) are used to map the different parameters to different styles. The rest components in the network are used to learn content information, which is shared by different styles.

4 PROBLEM DEFINITION AND SCOPE

4.1 Problem Definition

To design a real-time neural style transfer system which generates a stylized image from a content image and style image.

4.2 Scope

The scope of this project is to develop an algorithm which allows the generation of new images of high perceptual quality that combine the content of an arbitrary photograph with the appearance of numerous well-known artworks.Based on the implementation of [1,4,6] the style transfer of the content and style images can be done in real-time.

The approach trains feed-forward convolutional networks to generate multiple samples of the same texture to transfer artistic style from a given image to any other image. The networks used are very light-weight but hundreds of times faster.

5 METHODOLOGY

5.1 Approach

Both papers made use of pretrained VGG19 network[22]. Neither paper used the fully connected layers.

5.1.1 A Neural Algorithm of Artistic Style[1]

The main insight in this algorithm is that the representations of style and content of an image are separable in convolutional neural network. This approach uses a pretrained VGG19 network[22]. The input is taken three images in the VGG19 network[22] at the same time, a content image **C**, a style image**S**, and an image that is initialized to white noise **I**. All three images are passed through the network at the same time.

Let layer_i denote a layer in the VGG19 network[22], and let $\text{layer}_{i(x)}$ denote the output of layer i that takes an input x. For content reconstruction, let I_i denote the result of running **I** through VGG19[22], stopping after the i^{th} layer, let C_i denote the result of running C through VGG19[22], stopping after the i^{th} layer.

Pick any layer of the VGG19 net[22], we can use the output of that layer as the content representation. Suppose we picked layer i, we calculate the content loss by compute the MSELoss between I_i and C_i . For style representation, let S_i denote the result of running S through VGG19[22], stopping after the i^{th} layer. To represent the style, we need to use something called the Gram Matrix. The Gram matrix captures the correlation between different pixels in an image, while ignoring the specific details(content) of the image, thus making it a good candidate to represent style.

Let G denote the Gram matrix(for the definition of Gram matrix, please refer to [2]), the style representation of style image using the i^{th} layer be $G(S_i)$. The style loss is calculated by computing the MSELoss between $G(I_i)$ and $G(S_i)$. In the original paper, they used sum of style losses from multiple layers of VGG19[22] as the final style loss. Furthermore, we specify a content weight, C_W , and a style weight, S_W.

$$Total_{loss} = C_W * content_{loss} + S_W * content_{loss} \tag{1}$$

For each iteration, we try to minimize the total loss function by optimizing the values of **I**.

5.1.2 Real-time Neural Style Transfer[4,6]

The implementation is a combination from the concepts elaborated in Perceptual losses for real-time style transfer and super-resolution[4] and Instance Normalization: The Missing Ingredient for Fast Stylization[6] .The network architecture used is(Taken directly from the original paper[4]):

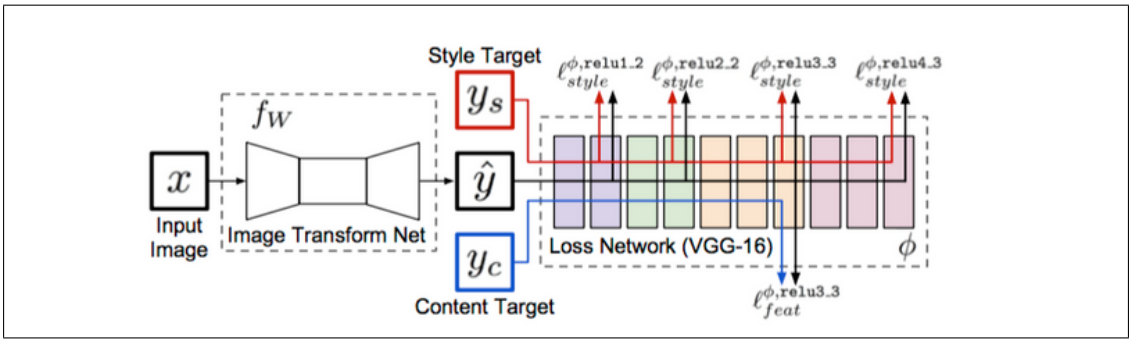

Figure 1: System overview. An image transformation network is trained to transform input images into output images.We use a loss network pretrained for image classification to define perceptual loss functions that measure perceptual differences in content and style between images. The loss network remains fixed during the training process.

x, the input image to the image transformation net, is the same as the content target. They pass y_s , $y\hat{}$, and y_c through VGG16 net[22] and calculate the style and content loss as in [2]. The idea is to have the image transformation net output a good $y\hat{}$ that can minimize the total loss function in the previous section. For the training set, COCO Dataset [4] is used as the content images and a single style image. In this approach, they have to train a separate network for every style image.

Layer	Activation size
Input	3*256*256
Reflection Padding(40*40)	3*336*336
32*9*9 conv,stride 1	32*336*336
64*3*3 conv,stride 2	64*168*168
128*3*3 conv,stride 2	128*84*84
Residual block,128 filters	128*80*80
Residual block,128 filters	128*76*76
Residual block,128 filters	128*72*72
Residual block,128 filters	128*68*68
Residual block,128 filters	128*64*64
64*3*3 conv,stride 1/2	64*128*128
32*3*3 conv,stride 1/2	32*256*256
3*9*9 conv,stride 1	3*256*256

Table 2: Layers and activation sizes of the network

As discussed in [6], replacing batch normalization with instance normalization significantly improves the quality of feed-forward style transfer models.These model described uses half number of filters per layer and with instance normalization instead of batch normalization. Using narrower layers makes the models smaller and faster without sacrificing model quality.

6 DATA COLLECTION

6.1 A Neural Algorithm of Artistic Style

There is no need of a training set since we are not training any network(we are essentially training the initialized image). Here are some examples of the images collected:

Figure 2: Content images used (From left to right): Golden Gate Bridge, Chicago skyline and Stata Centre,MIT

Figure 3: Famous paintings used as style images (From left to right): The Shipwreck of the Minotaur, The Great Wave off Kanagawa and Mysterious Rain Princess

6.2 Real-time Neural Style Transfer

For replicating this paper, we need the COCO dataset [21]. We first downloaded the COCO dataset [21], then we removed all non-RGB images, and resized all images in the dataset to 256 by 256. We used the same set of style images.

Figure 4: Famous paintings continued(From left to right): Udnie (Young American Girl, The Dance) and The Scream

7 IMPLEMENTATION DETAILS

7.1 A neural algorithm of artistic style

The hyperparameters in this algorithm include content weight, denoted as C_W , style weight, denoted as S_W , number of iterations trained, denoted as Ite, layers used to represent content, denoted as l_C, and the layers used to represent style, denoted as l_C.

Consider the below images as content image and style image respectively. It took approximately 2 hours 45 minutes to develop the stylized image on

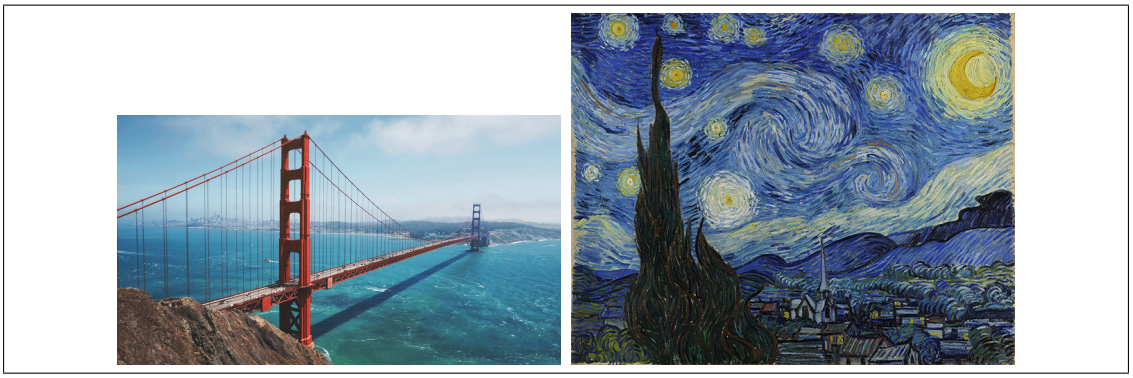

Figure 5: Content image: Golden Gate Bridge and Style image: Starry Night by Vincent Van Gogh

GeForce 940MX. The output image is initialized by a random image which over the course of iterations takes solid features from the content image and the while the texture of the style image is retained. As it can be seen from the image below the quality of the image generally increases with the number of iterations. But due to time and resource constraints, the output image is generated with only 1000 iterations.

The other results generated from this algorithm are:

Figure 6: The changes in the image over the course of iterations. The image is initialised with random noise. Each image is recorded after 200 iterations.(From left to right):After 0 iterations, After 200 iterations, After 400 iterations

Figure 7: Continued (From left to right):After 600 iterations, After 800 iterations, After 1000 iterations

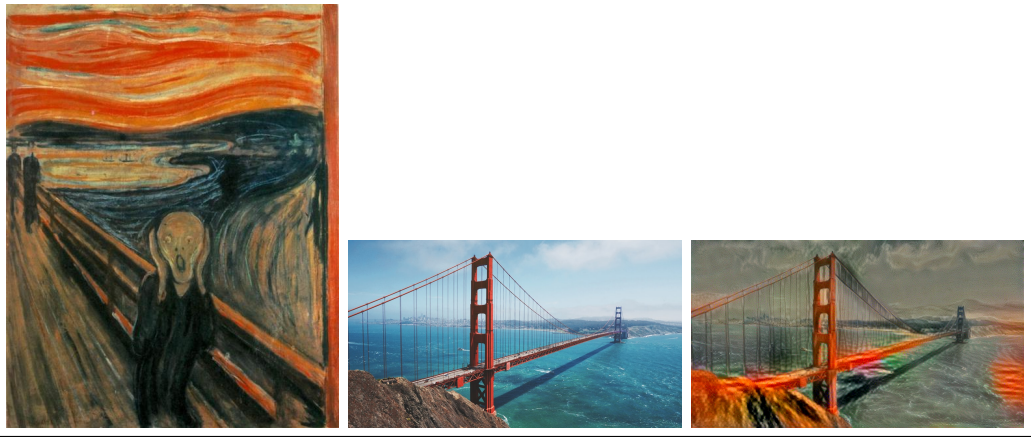

Figure 8: From left to right:Style,Content,Output

Figure 9: From left to right:Style,Content,Output

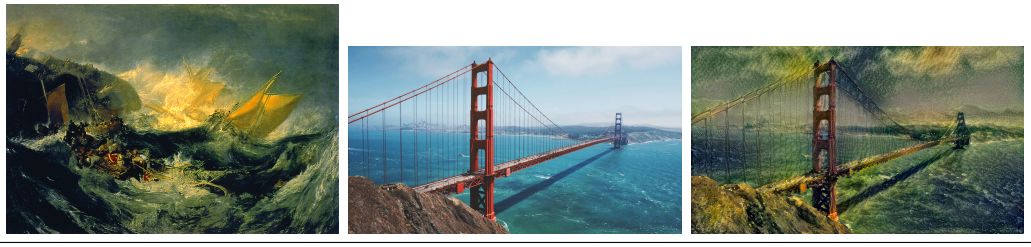

Figure 10: From left to right:Style,Content,Output

7.2 Real-time Style Transfer

During training, this algorithm takes 80000 content images from the COCO dataset as training set and just one style image. The algorithm looks at this one style image repeatedly. Therefore, it has a tendency to overfit to the style image. Thus at an instance the model is trained to develop stylized image for only one type of style image. It processes the input images to generate the stylized output images. Because it does not require any training, it generates the output in a single pass. It is just a matter of few seconds which drastically improves the efficiency. Furthermore, the use of instance normalization makes the network lightweight and improves the computational efficiency. Results generated from this algorithm are:

Figure 11: From left to right:Style,Content,Output

Figure 12: From left to right:Style,Content,Output

Figure 13: From left to right:Style,Content,Output

7.3 Comparison between the two algorithms

While the second algorithm generates output thousand times faster than the first algorithm, it it also retains the fine details and edges because it overfits to the

Figure 14: From left to right:Style,Content,Output

style image. Hyperparameter tuning for the real-time transfer algorithm is much harder because it takes a very long time to train the neural network from scratch, thus taking a very long time to try different sets of hyperparameters. And when the network is trained, it is only applicable for a specific style image. If you want to transfer a few styles to many content images, the real-time transfer algorithm will be a better choice. It clearly shows that the finer details are maintained using

Figure 15: Image generated using first algorithm taken 2 hrs 45 mins

real-time style transfer and it is also computationally efficient. However in matters concerning art, one can never say which of the two is a more successful style transfer. Which image is more accurate is subjective to every person's opinion.

Figure 16: Image generated using first algorithm taken 3s

8 CONCLUSION AND FUTURE ENHANCEMENTS

8.1 Conclusion

Through this project we have observed the applications of Convolutional Neural Networks in style transfer. We have demonstrated style transfer using both optimization-based methods and stand-alone feed-forward networks with perceptual loss functions. The use of feed-forward networks has shown drastic improvement in the quality of images by retaining finer details and has also reduced the time required to seconds. Also it can be concluded that, what constitutes a successful style transfer is not definitive.

8.2 Future Enhancements

One clear area of improvement is making the algorithm robust to a greater range of style and content images. This can be done by defining regions of the style image with a richer selection of style and also having a better segmentation process or removing the need for a mask in the algorithm. Another area of improvement is combining multiple styles from different images to stylize an image.

References

[1] Gatys, Leon A., Ecker, Alexander S., and Bethge, Matthias. A neural algorithm of artistic style. CoRR, abs/1508.06576, 2015b.

[2] Gatys, Leon, Ecker, Alexander S, and Bethge, Matthias. Texture synthesis using convolutional neural networks. In Advances in Neural Information Processing Systems, NIPS, pp. 262–270, 2015a.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In Advances in Neural Informa- tion Processing Systems (NIPS), pages 2672–2680, 2014.

[4] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings,Part II, pages 694–711, 2016.

[5] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR, abs/1511.06434, 2015.

[6] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, pages 1349–1357, 2016.

[7] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5188–5196.

[8] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2479–2486.

[9] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stylebank: An explicit representation for neural image style transfer," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1897–1906.

[10] Y. Li, F. Chen, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Diversified texture synthesis with feed-forward networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3920–3928.

[11] H. Zhang and K. Dana, "Multi-style generative network for real-time transfer," arXiv preprint arXiv:1703.06953, 2017.

[12] J. Johnson, "neural-style," https://github.com/jcjohnson/ neural-style, 2015.

[13] "DeepArt," 2016.: https://deepart.io/

[14] Luan, F., Paris, S., Shechtman, E., Bala, K.: Deep photo style transfer. In: CVPR. (2017)

[15] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz, “A closed-form solution to photorealistic image stylization,” arXiv:1802.06474, 2018.

[16] ”Fast Style Transfer”, https://github.com/lengstrom/fast-style-transfer

[17] ”Neural Style Transfer: A Review ”, https://github.com/ycjing/Neural-Style-Transfer-Papers.

[18] https://github.com/DmitryUlyanov/texture_nets

[19] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, Mingli Song, ”Neural style transfer: A review”, 2017.

[20] Tian Qi Chen, Mark Schmidt, ”Fast patch-based style transfer of arbitrary style”, arXiv preprint arXiv:1612.04337, 2016.

[21] ”COCO Dataset”, http://images.cocodataset.org/zips/train2014.zip

[22] ”VGG-19 pretrained network”, http://www.vlfeat.org/matconvnet/models/beta16/imagenet-vgg-verydeep-19.mat