Generating Stylistic Images by Extending Neural Style Transfer Method

Eisha Patel and Sri Krishnan
Department of Electrical, Computer, and Biomedical Engineering
Ryerson University
350 Victoria St, Toronto, ON M5B 2K3, Canada
(+1) 647-832-1336
eisha.patel28@gmail.com, krishnan@ryerson.ca

ABSTRACT

Fine arts have long been considered a reserved mastery for the minority of talented individuals in society. The ability to create paintings using unique visual components such as color, stroke, theme, and other creative aspects is currently beyond the reach of computer algorithms. However, there exist algorithms, which have the capability of imitating an artist's painting style and stamping it on to virtually any image to create a one-of-a-kind piece. This paper introduces the concept of using a convolutional neural network (ConvNet or CNN) to individually separate and recombine the style and content of arbitrary images to generate perceptually striking "art" [2]. Given a content and style image as reference, a pre-trained VGG-16 ConvNet can extract feature maps from various layers. Feature maps hold semantic information about both reference images. Loss functions can be developed for content and style by minimizing the mean-square-error between the feature maps used. These loss functions can be additively combined and optimized to render a stylistic image [6]. This technique is called Neural Style Transfer (NST) originally proposed by Leon Gatys in his 2015 research paper, "A Neural Algorithm of Artistic Style". This research project attempts to replicate and improve upon the work done by Leon Gatys. The purpose of this research is to experiment using a variety of feature maps and optimizing the loss function to identify visually appealing results. A total variation loss factor is introduced to minimize pixilation and sharpen feature formation. Images generated have been assigned a Mean Opinion Score (MOS) by a group of non-bias individuals to affirm the attractiveness of the results.

ICIME 2017, October 9–11, 2017, Barcelona, Spain

ACM ISBN 978-1-4503-5337-3/17/10…$15.00
DOI: https://doi.org/10.1145/3149572.3149592

Keywords

Convolution Neural Network (ConvNet or CNN); feature maps; Neural Style Transfer (NST); ImageNet; VGG Net; MOS scores

1. INTRODUCTION

This section provides the details regarding the concepts and findings behind this research on the Artistic Neural Style Transfer technique. The following items will help the reader familiarize with the basic concepts/terminology used in this paper.

1) **Convolutional Neural Network (ConvNet/ CNN):** are a category of neural networks capable of image recognition and classification. The architecture consists of two main components; hidden layers and classification layers. Hidden layers consist of a series of convolutions and pooling operations, which are responsible for feature extraction. The final classification layers are responsible for assigning probabilities as to what the input image possibly is.
2) **Feature Maps:** Each layer in CNN applies a collection of image filters to extract chunks of data called feature maps. If you had a picture of a zebra, this is the part where the network would recognize its stripes, two ears, and four legs.
3) **ImageNet**: is a large visual database containing over 14 million images designed for use in visual object recognition research.
4) **VGG-16 Net:** is a specific CNN architecture named after the creators from the Oxford Visual Geometry Group. It was used to win the ILSVR (ImageNet) competition in 2014 and classifies images with 93.5% accuracy. A gold standard even today.

Convolutional Neural Networks (CNN or ConvNet) is a class of deep neural networks, popularly known for their image recognition and classification capabilities. To train/test a CNN model, an input image is passed through a series of hidden layers,

which are responsible for feature extraction. Basically, each layer pulls out information about the image in chunks of data called feature maps. The final classification layers are responsible for assigning probability scores of what the input image possibly is.

Neural Style Transfer (NST) is a unique application of ConvNets. Given a content image and style reference image, an artistic image can be generated such that the content image inherits the style of the reference image. The style and content can be separately modeled by using various layers of feature maps. Leon A. Gatys first introduced this concept in his 2015 research paper [2], "A Neural Algorithm of Artistic Style". Since then, applications of NST are commonly seen in photo editing software such as *Prisma, Snapchat* and *Instagram*. This research project attempts to replicate the work of Leon A. Gatys and synthesize perceptually meaningful images. We will also propose techniques to improve the perceptual quality of generated images.

2. LITERATURE REVIEW

The following section provides an overview of the research articles associated with the proposed research work.

1) **"A Neural Algorithm of Artistic Style [2]"** *(Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aug 2015)*. This paper serves as the base of the replication study. Leon A. Gatys was the first to propose that a convolution neural network (CNN) could be used to create artistic images of high perceptual quality. To be more specific, the system uses neural representations to separate and recombine the content and style of arbitrary images. Gatys work showed that a CNN, pre-trained for image classification, was capable of encoding information about the semantic and perceptual qualities of a given image. He proposed that his work provides an algorithmic understanding of how the human brain creates and perceives imagery.

2) **"Very Deep Convolutional Networks for Large-Scale Image Recognition [10]"** *(Karen Simonyan and Andrew Zisserman, Sep 2014)*. This paper investigates the effect of convolutional network depth on the accuracy of large-scale image recognition. Simonyan and Zisserman performed a complete evaluation of networks of various depths using a 3x3 convolutional filter and the ImageNet database. Their models for the VGG-16 and VGG-19 architectures proved to achieve state-of-the-art results on any general dataset. These 2 architectures are publically available and the VGG-19 model was even used by Gatys in his paper "A Neural Algorithm of Artistic Style". I will be using the open-source VGG-16 network available in *Python* via *Keras*.

3) **"Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis [12]"** *(Li and Wand, Jan 2016)*. Li and Wand implement a linear combination of the style and content as a loss function along with a regularizing term. Often there is a significant amount of low-level image information lost during network training, which is why the reconstructed image looks noisy and unnatural. The regularizing term fixes this issue by adding smoothness to the final image. This compensating technique will be used to improve the perceptual quality of the generated images in our research as well.

4) **"Perceptual Losses for Real-Time Style Transfer and Super-Resolution [11]"** *(Johnson, et al, Mar 2016)*. Traditional methods of feed-forward convolution neural networks measure the per-pixel loss between the input and output images. Johnson proposes that we focus more on perceptual loss rather than per-pixel loss. Perceptual loss functions look at the high-level image features allowing us to reconstruct finer detailed images. In addition, the algorithm also performs significantly faster.

3. METHODOLOGY

Traditional ConvNets, trained for image recognition, learn to identify different graphical aspects along the hierarchy of layers. Hence higher layers tend to preserve only high-level content such as the arrangement of objects in the image, but pixel detail is progressively compromised [2]. The layer chosen for content reconstruction is completely subjective depending on how much we want to preserve the original image. To obtain an accurate representation of the style, we need to capture the colors and texture of the style reference image. This can be done by finding the correlation between multiple layers such that reoccurring details such as color and texture are captured while the global arrangement of objects in the image are not [2]. VGG-16 is a particular type of ConvNet that contains 13 convolution layers and 5 max pooling layers (Figure 3.1). We will be using this ConvNet for our application. The final fully connected (dense) layers are irrelevant, as we are not trying to classify the image. We only need the intermediate convolution layers to synthesize content and style. *Python* has set of built-in libraries and APIs that allow for us to easily implement neural style transfer. *Keras* is an open-source neural network library that runs on top of *Tensorflow*. We accessed our VGG-16 network from this library.

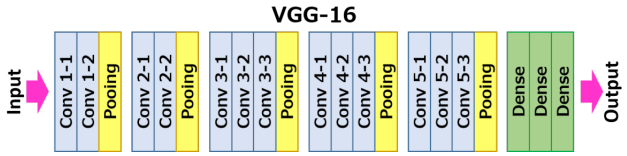

Figure 3.1: Architecture of a VGG-16 ConvNet [3]

At each stage of the CNN, a given input image can be represented by a set of filtered images. The number of filters applied to an image increases along the hierarchy, but the filter size decreases due to max-pooling [2]. ***Content Reconstruction:*** Content can be represented using any single layer. In Figure 3.2 the input content image is reconstructed at each layer of the network ((A) conv1_1, (B) conv2_1, (C) conv3_1, (D) conv4_1, (E) con5_1). Reconstructions made using the shallow layers are almost identical to the original image.

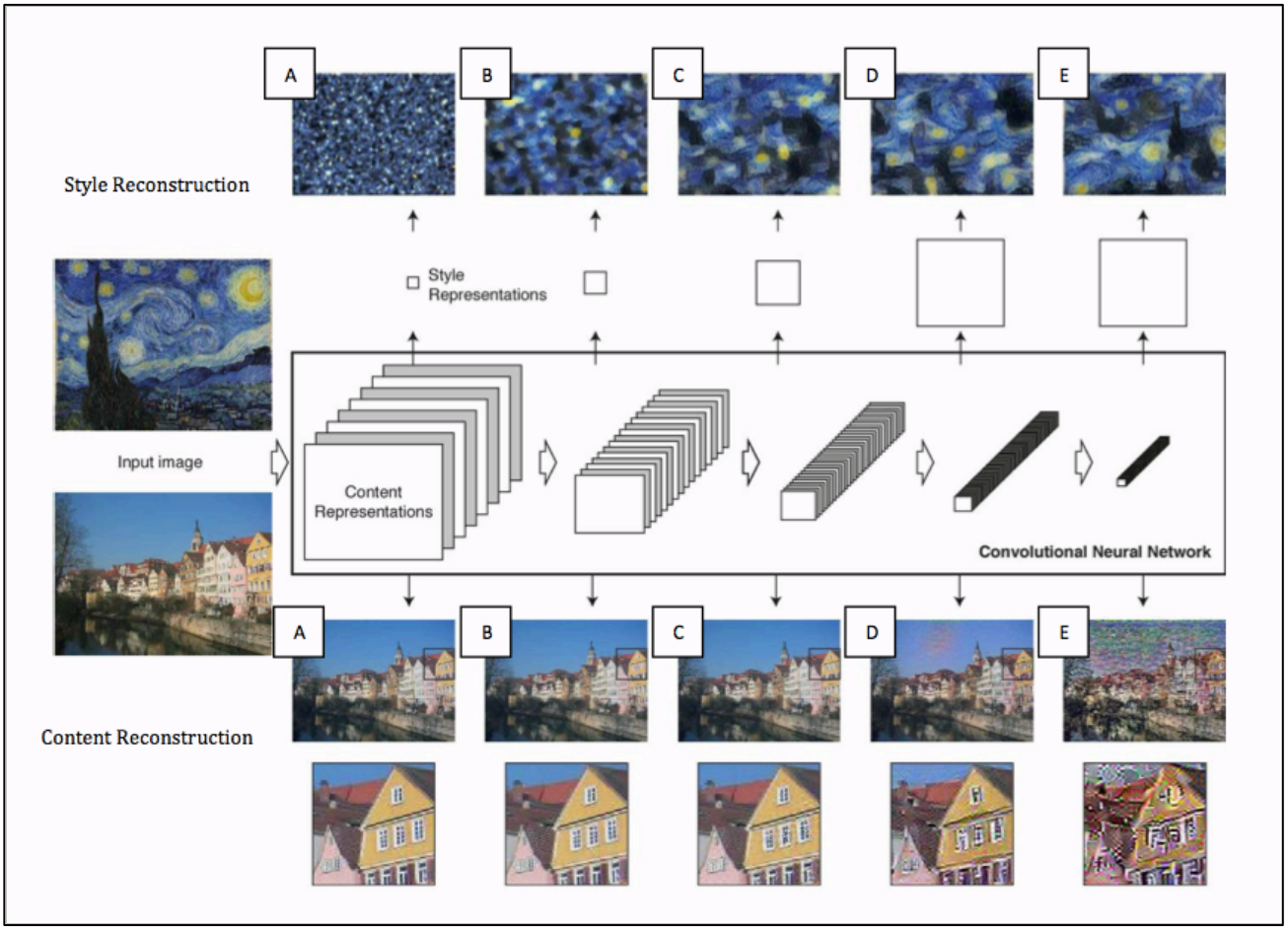

Figure 3.2: Reconstructing style and content using a convolutional neural network (image from Gatys et al. 2015)

Reconstructions made using the deeper layers are more pixilated but the high-level arrangement of objects is still preserved. ***Style Reconstruction:*** Style can be represented using a subset of layers. In Figure 3.2 the style of the input style image is extracted using permutations of layers. ((A) conv1_1, (B) conv1_1 + conv2_1, (C) conv1_1 + conv2_1 + conv3_1, (D) conv1_1 + conv2_1 + conv3_1 + conv4_1, (E) conv1_1 + conv2_1 + conv3_1 + conv4_1 + conv5_1). The style reconstructed using a larger subset of layers yields a more accurate representation of the style in terms of color and texture.

Given the required layers, we now know how to extract style and content individually. But how do we ensure that the generated image (G) only inherits the content of the content image (C) and not the style? Similarly, how do we ensure that the generated image (G) only inherits the style of style image (S) and not the content? These questions can be solved in terms of an optimization problem in which we minimize the loss between the generated image with respect to the content and style image. The loss function can be divided into two components; one for content loss and one for style loss [3].

$$L_{\text{total}}(S,C,G) = \alpha L_{\text{content}}(C,G) + \beta L_{\text{style}}(S,G) \text{ (Eq. 1)}$$

$$L_{\text{content}}(C,G) \approx 0 \textbf{ (Eq. 2)}$$

$$L_{\text{style}}(S,G) \approx 0 \textbf{ (Eq. 3)}$$

We begin by initializing the generated image as a matrix of random white noise. The goal is to iteratively minimize both counter-parts (Eq. 2 and Eq. 3) such that the features of the generated image progressively resemble the style and content better. During each iteration, all three images (C, S, and G) are passed through the VGG-16 network [3]. The activation values, which encode feature representations of the given image at a particular layer, are taken as inputs for the loss function. The hyper parameters α and β in Eq. 1 are "control knobs" which allow us to dictate how much content/style we want the generated image to inherit [6].

To compute the content loss, we pass the content image (C) and generated image (G) at a particular layer, say conv3_3, of the VGG-16 and retrieve the activation values. We then find the Euclidean Norm element-wise between the activation values of image C and G. The mathematical form of content loss is given by Eq. 4. Let $\boldsymbol{L}$ represent the layer whose activation outputs we

use to derive the content loss. The activation layer of the content image is denoted as ***a[L](C)*** and the activation layer of the generated image is denoted as ***a[L](G)*** [7].

$$L_{\text{content}}(C,G,L) = \frac{1}{2}\sum_{ij}(a[L](C)_{ij} - a[L](G)_{ij})^2 \text{ (Eq. 4)}$$

A single layer in the VGG-16 network consists of multiple feature maps. We need to find the correlation between the activations across all feature maps of the same layer. Hence, if feature map 'A' activates upon seeing a ball and feature map 'B', of the same layer, activates upon seeing a wheel, then they are likely to be correlated by shape. Such correlated patterns/textures/colors can be detected using the Gram Matrix [7]. Let ***GM[L](S)*** denote the Gram Matrix of the style image at layer ***L*** and ***GM[L](G)*** denote the Gram Matrix of the generated image at layer ***L.*** Now it's just a matter of finding the Euclidean Norm between the two matrices and minimizing their differences (Eq. 5). N_l is the number of feature maps in layer l and M_l refers to the dimensions of a feature map in layer l [7]. Computing the style loss requires more work because we are dealing with a subset of multiple activation layers. We can apply a weighted sum of all layers to calculate the total style loss (Eq. 6)

$$L_{\text{GM}}(S,G,l) = \frac{1}{4N_l^2 M_l^2}\sum_{ij}(GM[l](S)_{ij} - GM[l](G)_{ij})^2 \textbf{ (Eq. 5)}$$

$$L_{\text{style}}(S,G) = \sum_{l=0}^{L} w_l * L_{GM}(S,G,l) \textbf{ (Eq. 6)}$$

Now that we have Eq.'s. 4 and 6 for content and style loss respectively, we just need to add them up as per Eq. 1. Then we can implement any optimizer to perform gradient descent and iteratively reduce the loss in order to obtain a perceptually meaningful image [6]. A total of 12 iterations yield a stable image as the loss value approaches a global minimum. Specifications regarding the packages/APIs used, image pre-processing, and gradient optimization can be found in the *Python* code link (see appendix).

4. RESULTS

Sets of three experiments were conducted and the individual results for each are detailed in the sections below.

4.1 Experiment 1

In order to generate a "perfect" blend of images, we must experiment using a variety of network layers while varying the hyper parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. In Figure 4.1 we have a photograph of the iconic *Lana Del Rey* as our content reference and the famous painting "*Starry Night"* by *Vincent van Gogh* as our style reference. The goal is to see how varying parameters affect the generated image. In order to extract the style, we have used the following subset of layers; conv1_1, conv2_1, conv3_1, conv4_1, and conv5_1. By using the first layer of each block in the VGG- 16 network, we offer a fair sample of feature maps [9] (see Figure 3.2). This subset of style layers will remain consistent through all three experiments. The only layer we will be varying is the layer chosen for content extraction.

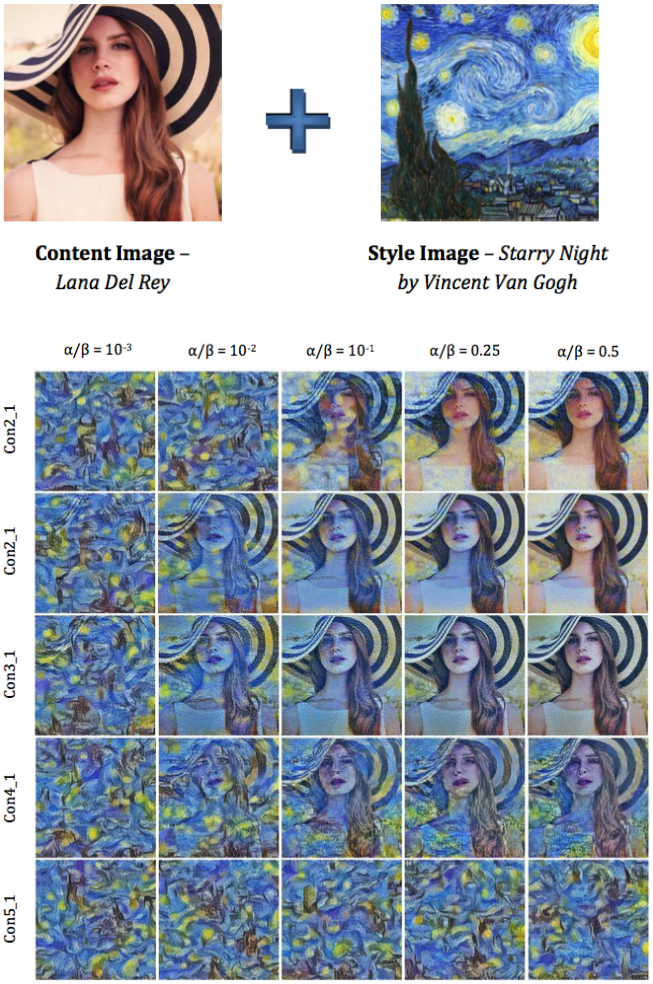

Figure 4.1: Detailed breakdown of layers against α/β. As the ratio of α/β increase, there is a higher emphasis on content. As the ratio of α/β decreases, there is a higher emphasis on style. Layer 3 (con3_1) shows the most balanced inheritance of style and content. Images generated using shallow layers of the network closely resemble the original content image. Images generated using deeper layers of the network offer a more abstract look. These findings align with the work of Leon A. Gatys [9]

4.2 Experiment 2

Recall, the loss function allows us to control the amount of content and style we want a generated image to inherit. A higher α/β ratio will yield an output image more representative of the original target image, while the opposite will yield an output image with stronger stylistic features. We can apply an optimizer on the total loss (Eq. 1) to perform gradient descent and iteratively decrease the loss. However, the results generated using just content and style loss are unappealing and pixilated. The noise in a generated image can be compensated by adding an additional term called total variation loss (TVL) to Eq.1. TVL looks at the sum of absolute differences between neighboring

pixels within an image and tries to minimize it [8]. This encourages image consistency, minimized pixilation and sharper feature formation [8]. Assuming that x_{ij} represents the pixel value at coordinate (i,j), and so the total variation loss can be written as:

$$\sum_{i,j}|x_{i,j} - x_{i+1,j}| + |x_{i,j} - x_{i,j+1}| \textbf{ (Eq. 7)}$$

Hence, the total loss function is now a weighted sum of content, style, and total variation loss. Figure 4.2 shows the before/after effects of adding TVL to the total loss. There is a noticeable reduction in the grainy texture (noise) of the after TVL results. In addition, style adaptation is more prominently inherited as we see more texture being integrated.

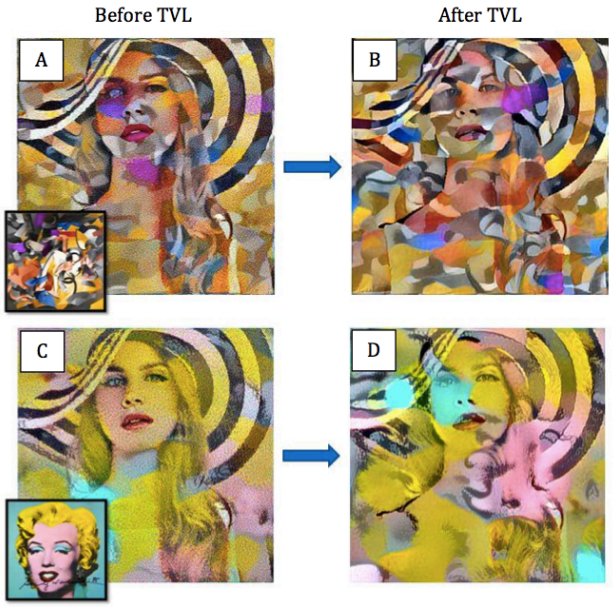

Figure 4.2: Before and After Results of TVL. Images A, B, C, and D were created using the following parameters and layers; α/β = 0.01, style layers = conv1_1, conv2_1, conv3_1, conv4_1, and conv5_1, content layer = conv2_2. The weight given to TVL loss in image B and D is 1.0.

4.3 Experiment 3

NST is truly a remarkable fusion of artificial intelligence and the arts. A neural network is completely blind to all the variances that preserve the identity of a subject with an image, yet it is beautifully able to extract the content/style. This makes NST highly versatile to a variety of visual subjects. Experiment 3 is an attempt to play *wildcards* with a variety of reference images purely for curiosity (Figures 4.3 and 4.4).

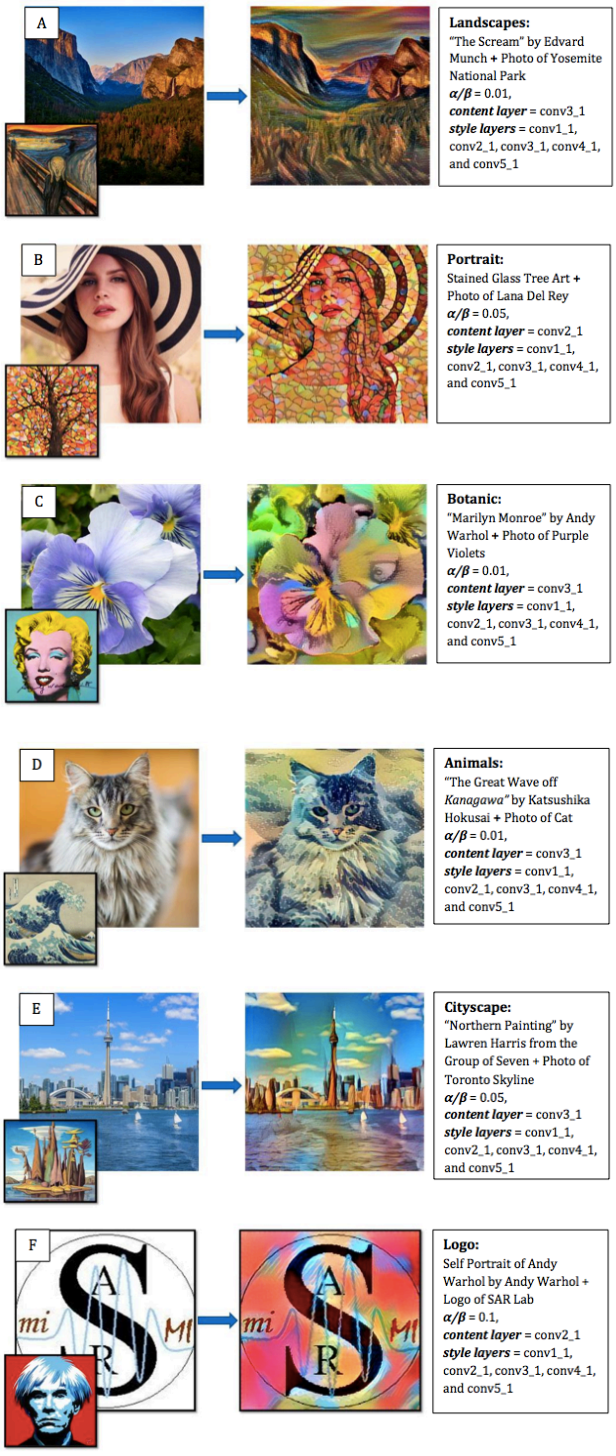

Figure 4.3: Results of NST using a Variety of Subject Matters. Content and style references are shown on the left-hand side and the generated image is shown on the right-hand side. Details regarding the image and parameters used to create them are also on the right.

Now what would happen if we were to use a painting as our content reference and a photograph as our style reference? Figure 4.4 below shows an example of exactly that.

Figure 4.4: Reverse Content and Style. A photograph has been used as style reference and a painting has been used as content reference. Typically most NST creations occur in the vice versa order. Details regarding the image and parameters used to create them are on the right-hand side.

To evaluate the perceptual quality and attractiveness of these images would be highly subjective from individual to individual. The mean opinion score (MOS) test is an arithmetic mean of all the ratings collected from a subjective quality evaluation test. Hence a MOS test was conducted in survey format to judge these images. A group of 45 selected individuals were asked to anonymously rate the generated images from a scale of 1 to 5 (1 = bad, 2 = poor, 3 = fair, 4 = good, 5 = excellent). All scores have averaged and summarized in the table below (Table 4.1).

Table 4.1: MOS test summary.

Image	MOS
Fig 4.3. A	4.0
Fig 4.3. B	4.37
Fig 4.3. C	3.37
Fig 4.3. D	4.33
Fig 4.3. E	4.0
Fig 4.3. F	3.73
Fig 4.4. G	3.66

Based on the mean opinion scores (MOS), we can conclude that most individuals found the images aesthetically pleasing. Note that individuals surveyed were not particularly artists or AI specialists. Participants came from a variety of backgrounds including health care, business, engineering, social sciences, computer science, etc. Images synthesized using reference images with similar color schemes (example image A), blend well together because the algorithm attempts to match the areas where the color contrast is similar (example: blue skies above the mountains adapt the colors of the blue water-body in the painting). Similar patterns can be noticed with texture (example image D: the elongated fur of the cat matches the flow of lines in the waves, hence the fur is replaced with strokes of waves) and shape (example image E: tall buildings in the skyline inherit the shape of the tall trees in the painting). Such similarities in color, texture, and shapes between style/content reference images result in successful blends (MOS scores >= 4.0). Also note that when working with content images containing fine details such as facial features (example image B) and text (example image F), it is ideal to use shallow layers of the VGG-16 network for content reconstruction in order to preserve such details.

5. CONCLUSION/ FUTURE WORK

The goal of this research was to replicate and extend the work of Leon Gatys' 2015 research paper, "*A Neural Algorithm of Artistic Style*". More specifically, we wanted to appreciate the beauty of convolutional networks and their ability to separate the style and content of arbitrary images without any understanding of the image context. Our research findings from Experiment 1 confirm the work done by Leon Gatys by showing a similar relationship between network layers and hyper parameters. In Experiment 2 we showed that by adding an additional TVL term to the loss function, we can easily reduce the "graininess" of a generated image. This technique is an extension to what was originally proposed by Leon Gatys in 2015. For Experiment 3, we applied NST on a variety of pictures and performed a MOS subjective perceptual test to evaluate the visual attractiveness of each generated image. All ratings fell between the range of 3.37 to 4.37, which indicates that the images generated were on average fair to good quality.

Many improvements have been made to the traditional methods of NST proposed back in 2015. Despite the remarkable results we witnessed, implementing NST is a very slow iterative process. The image is optimized using back propagation until the target results are achieved. *Fast Neural Style Transfer* (FNST) is a quicker version of the traditional NST, which eliminates the need for iterations and generates an image in a single pass, allowing us to metamorphose an entire video in real-time [8]. Traditional NST use *per-pixel* loss functions but FNST defines optimizing functions that look at the *perceptual* loss based on high-level features extracted from the network. By optimizing perceptual features rather than pixels, we can generate higher quality images in less time [4].

6. ACKNOWLEDGMENTS

We gratefully acknowledge the insights provided by the volunteers who performed the MOS test. The support of Signal Analysis Research (SAR) group and Ryerson University is also appreciated.

7. REFERENCES

[1.] Champandard, Alex J. "'Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artwork.'" 5 Mar. 2016, https://github.com/LuckyZXL2016/Deep-Learning-Papers-Reading- Roadmap/blob/master/3.7-Art/Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artwork.pdf.

[2.] Gatys, Leon, et al. "A Neural Algorithm of Artistic Style." 1 Sept. 2016, https://arxiv.org/abs/1508.06576

[3.] Hnarayanan. "Hnarayanan/Artistic-Style-Transfer." *GitHub*, https://github.com/hnarayanan/artistic-style-transfer/blob/master/notebooks/6_Artistic_style_transfer_with_a_repurposed_VGG_Net_16.ipy

[4.] Johnson, Justin, et al. "Perceptual Losses for Real-Time Style Transfer and Super-Resolution." 27 Mar. 2016, https://arxiv.org/pdf/1603.08155.pdf.

[5.] Mahendran, Aravindh, and Andrea Vedaldi. "Visualizing Deep Convolutional Neural Networks Using Natural Pre-Images." 14 Apr. 2016, https://arxiv.org/pdf/1512.02017.pdf.

[6.] Narayanan, Harish. "Convolutional Neural Networks for Artistic Style Transfer." *Convolutional Neural Networks for Artistic Style Transfer - Harish Narayanan*, https://harishnarayanan.org/writing/artistic-style-transfer/.

[7.] "Neural Style Transfer: Creating Artificial Art with Deep Learning and Transfer Learning." *Packt Hub*, 22 Nov. 2018, https://hub.packtpub.com/neural-style-transfer-creating-artificial-art-with- deep-learning-and-transfer-learning/.

[8.] Ulyanov, Dmitry, et al. "Instance Normalization: The Missing Ingredient for Fast Stylization." 6 Nov. 2017, https://arxiv.org/pdf/1607.08022.pdf.

[9.] William Falcon. "Accessible AI - A Neural Algorithm of Artistic Style." *William Falcon*, William Falcon, 3 Sept. 2017, https://www.williamfalcon.com/accessible-ai-blog/2017/9/3/a-neural- algorithm-of-artistic-style-transfer-summary.

[10.] Simonyan, K., & Zisserman, A. (2015, April 10). Very Deep Convolutional Networks for Large-Scale Image Recognition. Retrieved from https://arxiv.org/abs/1409.1556

[11.] Johnson, J., Alahi, A., & Fei-Fei, L. (2016, March 27). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. Retrieved from https://arxiv.org/abs/1603.08155

[12.] Li, C., & Wand, M. (2016, January 18). Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. Retrieved from https://arxiv.org/abs/1601.04589

8. APPENDIX

[1.] Code for NST: https://github.com/eispat28/GENERATING-STYLISTIC-IMAGES-USING- CONVOLUTIONAL-NEURAL-NETWORKS

[2.] Images used for NST: https://github.com/eispat28/GENERATING-STYLISTIC-IMAGES-USING- CONVOLUTIONAL-NEURAL-NETWORKS

[3.] ImageNet Dataset: http://www.image-net.org/

[4.] Documentation on Python Keras Tensorflow: https://www.datacamp.com/community/tutorials/cnn-tensorflow-python

Authors' background

Your Name	Title*	Research Field	Personal website
Eisha Patel	Former Data Science Masters' Student	Image recognition and processing	https://github.com/eispat28 https://www.linkedin.com/in/eishapatel/
Sri Krishnan	Professor	Signal and Image Analysis	www.ee.ryerson.ca/~krishnan

*This form helps us to understand your paper better; the form itself will not be published.

*Title can be chosen from: master student, Phd candidate, assistant professor, lecture, senior lecture, associate professor, full professor