Image Transfer Style in Neural Networks

Roberto Vasquez
Department of Computer Science
San Diego State University
robertovrios@gmail.com

Mariano Hernandez
Department of Computer Science
San Diego State University
mariano.hndz@gmail.com

James Foti
Department of Computer Science
San Diego State University
jameswilliamfoti@gmail.com

1. Abstract

There are copious amounts of applications that can merge two images together; however, not many that use machine learning in their techniques. An emerging technique in use today is the Artistic Style Transfer introduced by Leon Gatys. Artistic image style transfer uses machine learning algorithms to apply the style of one image onto another. Using machine learning techniques allows for the optimization problem at hand.

To achieve such extractions from style images we'll be using a Convolution Neural Network (CNN) with the Visual Geometry Group (VGG) architecture. This shall allow for the extraction and transfer of the style image onto the content image.

2. Introduction

This paper shall serve as a final term presentation as directed by the Machine Learning course at San Diego State University.

It will describe the topic of neural style transfer; previous models, what is artistic style transfer, the project implementation, the shortcomings of style transfer, model used along with results, and finally an assessment.

3. Image Style Transfer without Neural Networks

Previous to neural networks, transferring style image were developed by methods, such as, texture synthesis method which uses a statistical model to describe the image's texture, and a quantitative model to create a new color based on the main image color; or simplification and abstraction methods that combine a number of filters to render a given image. Additionally, methods such as stroke-based rendering (SBR) uses algorithms and discrete elements called strokes to generate desired styles into a given image resembling to real oil canvas.

4. Artistic Style Transfer

Artistic Style Transfer was first introduced by Leon Gatys in 2015. Since its discovery, Gatys used this technique as an optimization method that makes transferring a style possible on a given content image. As an optimization problem, it will be resolved using data training into a neural network; in other words, using two loss function: content and style loss that measures the difference between style and content images [2]. That is to say, the loss function tends to zero when its input images are similar to each other in terms of content and grows as their content deviates; and the style loss function tends to zero when its input images are similar to each other in terms of style and grows as their style deviates.

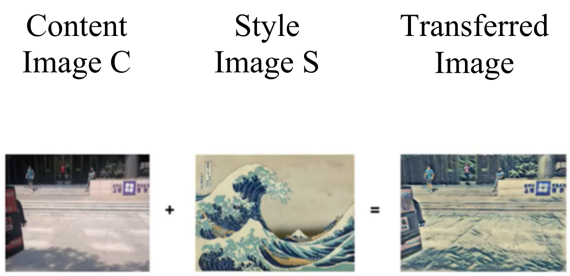


Figure 1. AST responds the problem statement: Given a content image C and a style image S, how can we apply the style of S on C in order to create a new image?

This paper explains how to pose and resolve this problem as an optimization problem using machine learning methods. Furthermore, this document shows results and conclusions of using a specific model.

5. The Style Transfer Problem

The style transfer problem could be defined as finding an image that differs minimum possible in content from the content image and differs minimum possible in style from the style image. Through training neural network, it is possible to minimize both the style and content loss function and find a solution to this transfer problem.

6. Convolutional Neural Network (CNN)

A convolutional neural network pre-trained for image classification already knows how to encode semantic and perceptual image information. Using a CNN as a basis to extract content and style representations from images.

7. VGG-19

Created by Simonyan and Zisserman, VGG-19 is a 19-layer CNN composed by 16 conv., 3 fully-connected (Fig. 2) and uses strictly used 3×3 convolution filters with stride and pad of 1, along with 2×2 max-pooling layers with stride 2. VGG-19 is a deeper CNN with more layers compared to the CNN in AlexNet. VGG-19 uses 3x3 convolutional filters in all layers to reduce the number of parameters and the error rate from 8.6% to 7.5% doing better than VGG-16. Moreover, it has been trained to classify 1000 object categories on more than a million of images.

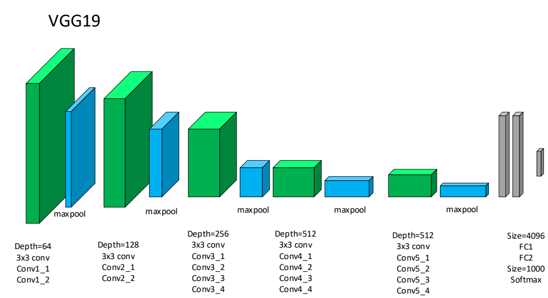


Figure 2. Network architecture of VGG-19 model: conv means convolution, FC means fully connected. *Image retrieved from "Very Deep Convolutional Networks for Large-Scale Image Recognition," Simonyan, K., Zisserman, A. (2014).*

8. Model

Our model uses the pre-trained CNN models that recognize objects within images as a basis to extract content and style representations from images. VGG-19 network architecture using intermediate layers makes enough to define and match the content and style representation of the input images. These representations will be independent from each other, so we can use the content representation from one image and style representation from another to generate a new image.

For our style transfer problem, we can use the fact that the activation maps or feature maps in a deeper convolution layers of the network represent larger scale features of the original image. Furthermore, if we recreate the original image from deeper layers, our model could keep exact level content but losing accuracy on the image pixel information. However, we are able to extract a style representation looking at the spatial correlation of values within an activation or future map.

The Gram matrix contains the correlation between every pair of feature map and captures a tendency of the features to co-occur in different parts of the image.

Gatys found that the best results are achieved by taking a combination of deep and shallow convolution layers as the style representation of an image.

$$G_{ij} = \sum_k F_{ik} F_{jk}$$

Posing Style Transfer as An Optimization Problem [1]:

$$\mathcal{L}_{content} = \frac{1}{2} \sum_{ij} \left(F_{ij}^{l} - P_{ij}^{l} \right)^2$$

We will now fully pose style transfer as an optimization problem. The first equation defines a content loss function. Giving a chosen content layer L, the constant loss is defined as the Euclidean distance between the feature map F in layer L of our content image C and the feature map P in layer L of our transferred image Y.

$$\mathcal{L}_{style} = \frac{1}{2} \sum_{l=0}^{L} \left(G_{ij}^{l} - A_{ij}^{l} \right)^2$$

The second equation defines the style loss function. Giving a choose layer L, the style loss is defined as the Euclidean distance between the Gram matrix G and layer L of our style image G and the Gram matrix A in layer L of our transferred image Y.

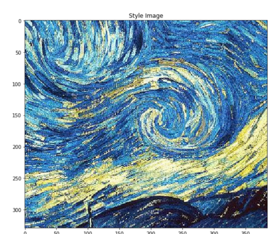
$$\mathcal{L}_{Total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style}$$

The third equation defines total loss which is simply a weighted sum of style and content losses, where the weights alpha and beta can be adjusted to allow the transfer image to contain either more content or style [3]. This total loss functions are we need to minimize to perform the task of style transfer.

9. Model Implementation

Our VGG-19 model has been built using tf.keras.applications.VGG19 library. To implement the model, we performed the following steps:

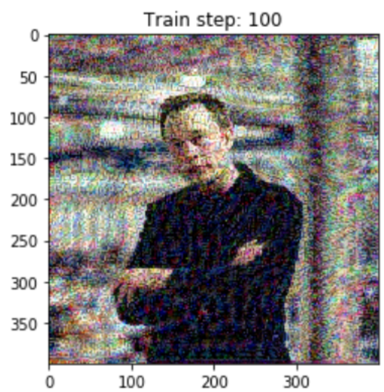
a. Upload content and style images.

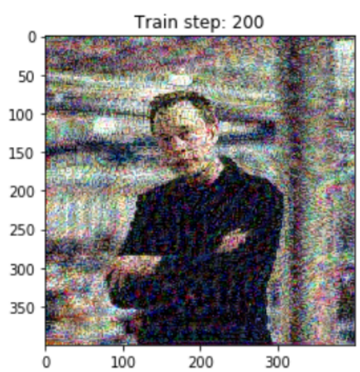


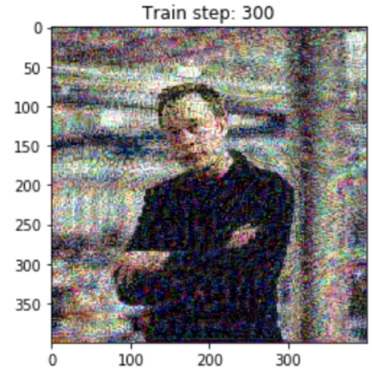
b. Reshape both content and style images into a specific shape and an array of numbers using tensor shape. As a result, both images have a shape of [1, 400, 400, 3].
c. Uploading the pretrained models VGG-19 in keras and create our model defining content and style layers.
d. Calculate the Gram matrix using the formulas defined by the CNN model.
e. Building a custom model VGG that is composed by the layers mentioned in the first step and which will be used to return the content and style features from respectively images.

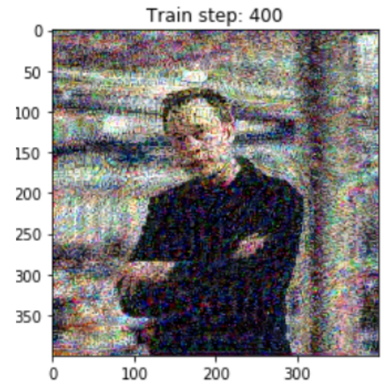
f. Optimize the model using the Adam algorithm that is a stochastic gradient descent method. We decided to use this method because it is included in Keras solution.
g. Define the custom weights for style and content images.
h. Calculate the loss function and gradient descent of the model.
i. Define the target image.
j. Execute the neural style transfer. It runs 1000 iterations (epochs = 10, steps = 100) appears to present good results without going overboard.

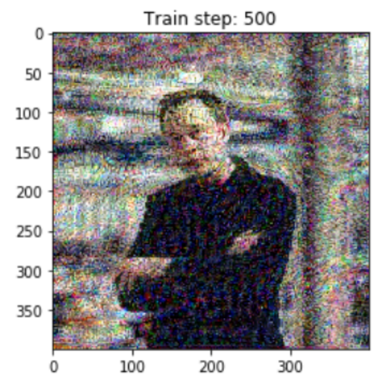
These are the results for the steps above:



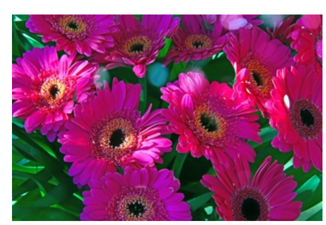


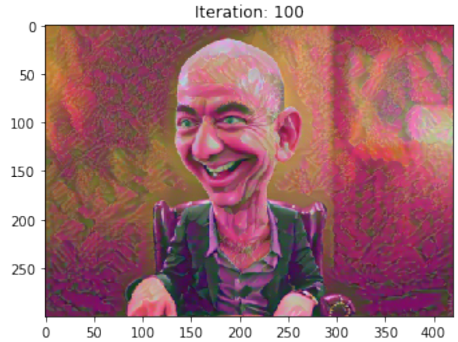


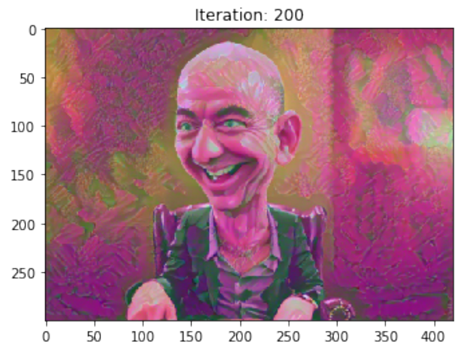


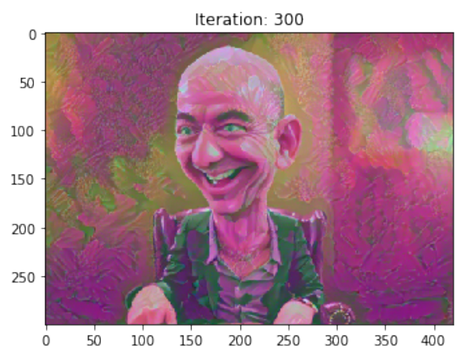


Now, an example using a Pytorch implementation given the following content and style images, respectively:

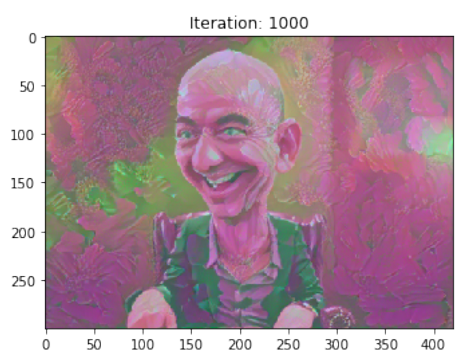


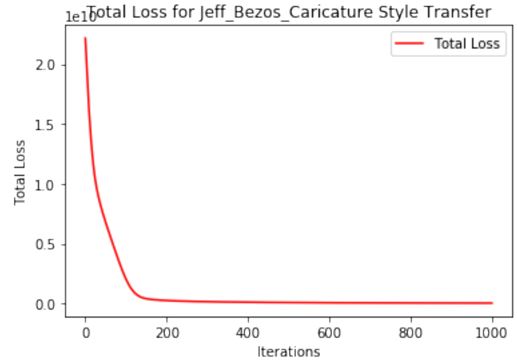






...





Here we can see the total loss based on the derivation of both the content and style losses from the distance in representational space of the VGG image classification network [2].

10. Application uses

Image transfer style have been applied in many fields specially in social media where users are able to share fantastic images. Style transfer image is also a creative art tool that have been used by artist to improve their work. Additionally, film and game industry also are using this technology to improve the process of creativity reducing costs and saving production time. However, there are still some drawbacks by using this technology. For example, there is a lack of process transparency that creates difficulties to control and achieve desired outputs. Moreover, it is struggled to balance flexibility, speed, and quality; thus, considering a quality evaluation module could be a good option to diminish an unacceptable output.

11. Conclusion

This paper is an experimental guide about Neural Style Transfer practice. Moreover, this paper explains an applied model in order to create new images using NST and style and content images. After the 5000 iteration, we can notice that the output images are totally distorted converging to the style input image. Furthermore, this paper demonstrates that it is possible to resolve the problem to find an image that differs minimum in content from the content image and differs minimum in style from the style image. Because NST is a relatively a new topic, it will be improved using features in an unsupervised manner. The development of these models could play an important role in photograph and video industry.

References:

[1] Gatys Leon A., Ecker Alexander S., Bethge Matthias. A Neural Algorithm of Artistic Style, 2015. Retrieve from https://arxiv.org/abs/1508.06576

[2] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. Proceedings of the British Machine Vision Conference (BMVC), 2017.

[3] Y. Li, T. Zhang, X. Han and Y. Qi, "Image Style Transfer in Deep Learning Networks," 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, 2018, pp. 660-664. doi: 10.1109/ICSAI.2018.8599501