Transfer Style with Keras

Jacobo Posada-Hoyos* A00829481

***Abstract*—In this project a implementation of a CNN with Neural Style Transfer is made to create new images based on different styles pictures by using VGG16 and VGG19 Convolutional Neural Network models for object recognition.**

***Index Terms*—Convolutional Neural Network, Deep Learning, Style Transfer, VGG16, VGG19.**

I. Introduction

THE developments and progress toward Machine Learning (ML) and specifically on Convolutional Neural Networks (CNN) have been one of the major topics of interest and developments in computational sciences due to their great potential in visual perception such as object and face recognition. Neural Style Transfer (NST) is an optimization technique to transfer the style information of one image to other image and is commonly used to create art or coloring B&W images [1].

II. Neural Style Transfer

The principle of NST is to define two distance functions (losses) to describe how different the content of two images are ($L_content$) and to describe the difference between two images in terms of their style (L_style). So, having three images, the content, the style and the output image, the algorithm minimize the content and style distances to create an output image that contains both contents.

III. Setup

In order to perform the style transfer model first is required make a pre-processing of the data constraining the images to be in a range of [0,255]. Then, define the pre-trained CNN model to extract the feature maps, in this case $VGG16$ and $VGG19$. Lastly, it is necessary to define the content and style loss functions and the backpropagation algorithm.

A. Content Loss

The content loss is define by the euclidean distance between the representation of both images (See Equation 1).

$$L^l_{content}(p,x) = \sum_{i.j}(F^l_{ij}(x) - P^l_{ij}(p))^2 \tag{1}$$

where the representation F^l_{ij} describe the input x and the representation P^l_{ij} describe the input p at the layer l. Both of those representation belong to the be the deep CNN denoted by $C_nn(X)$.

School of Engineering and Science,Tecnológico de Monterrey, Campus Querétaro, Ave. Epigmenio González 500, Fracc. San Pablo, Santiago de Querétaro, Querétaro, México 76130.
*E-mail: a00829481@itesm.mx

B. Style Loss

The style loss is described by the comparison between the Gram matrices that are the inner product of the vectors (i,j) to get the distance between the values and says if are close (more loss).

$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}(G^l_{ij} - A^l_{ij})^2$$

where G^l_{ij} and A^l_{ij} are the respective style representation in layer l of x and a. N_l is the number of feature maps of size $M_l = height * width$.

$$L_{style}(a,x) = \sum_{l \epsilon L}\frac{1}{|L|}E_l \tag{2}$$

IV. CNN Models

The VGG model is a convolutional neural network model proposed by K. Simonyan and A. Zisserman [2] used for localization and classification of objects. Two models were proposed for depths of 16 and 19 weight layers, the specification of the models are shown in the next subsections.

A. VGG16

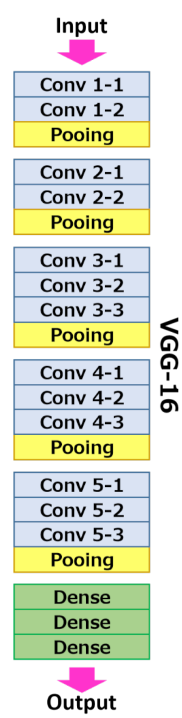


Figure 1: Structure of the VGG16 model

B. VGG19

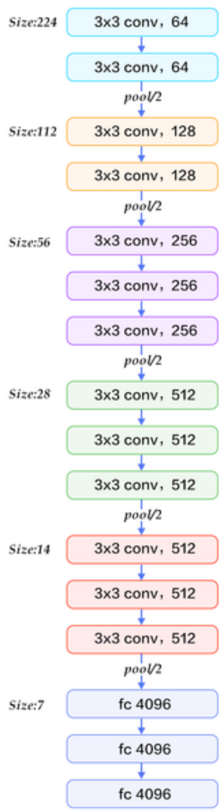


Figure 2: Structure of the VGG19 model

V. Experimental Results

Having the model and the loss functions the NSL process is implemented using a set of 8 images (4 styles, 4 contents) for each of the two models, $VGG16$ and $VGG19$.

For the optimization part, different values for the learning rate of the Adam Optimizer were used. For example, for the four groups the learning rate was of 0.03, 0.01, 0.5, 5, respectively. This is in order to see the evolution of learning as the learning rate changes as well as to compare their behavior with the pre-trained models. The results are shown in next page.

VI. Conclusion

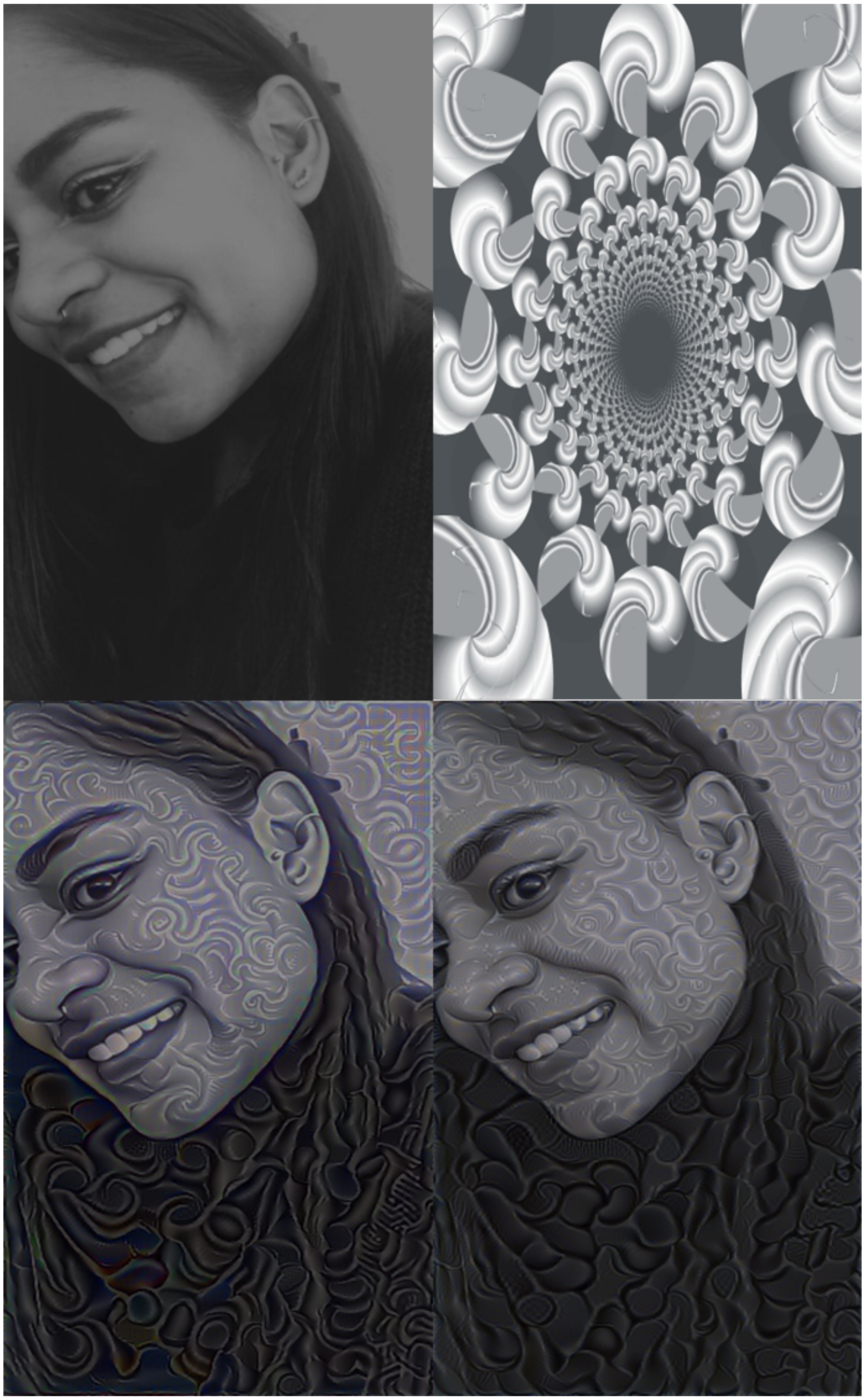
The results obtained in the Neural Style Transfer process demonstrated that it is a good technique for merging information between images.
Also, proved to be a great method for generating new content, a field that have a lot of potential, in addition to its powerful applications in image coloring and data augmentation. It was also found that the Adam Optimizer is a great tool to avoid problems with low gradients in noise problems.

References

[1] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. *A Neural Algorithm of Artistic Style*. 2015. arXiv: 1508.06576 `[cs.CV]`.

[2] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. arXiv: 1409.1556 `[cs.CV]`.

(a) (b)



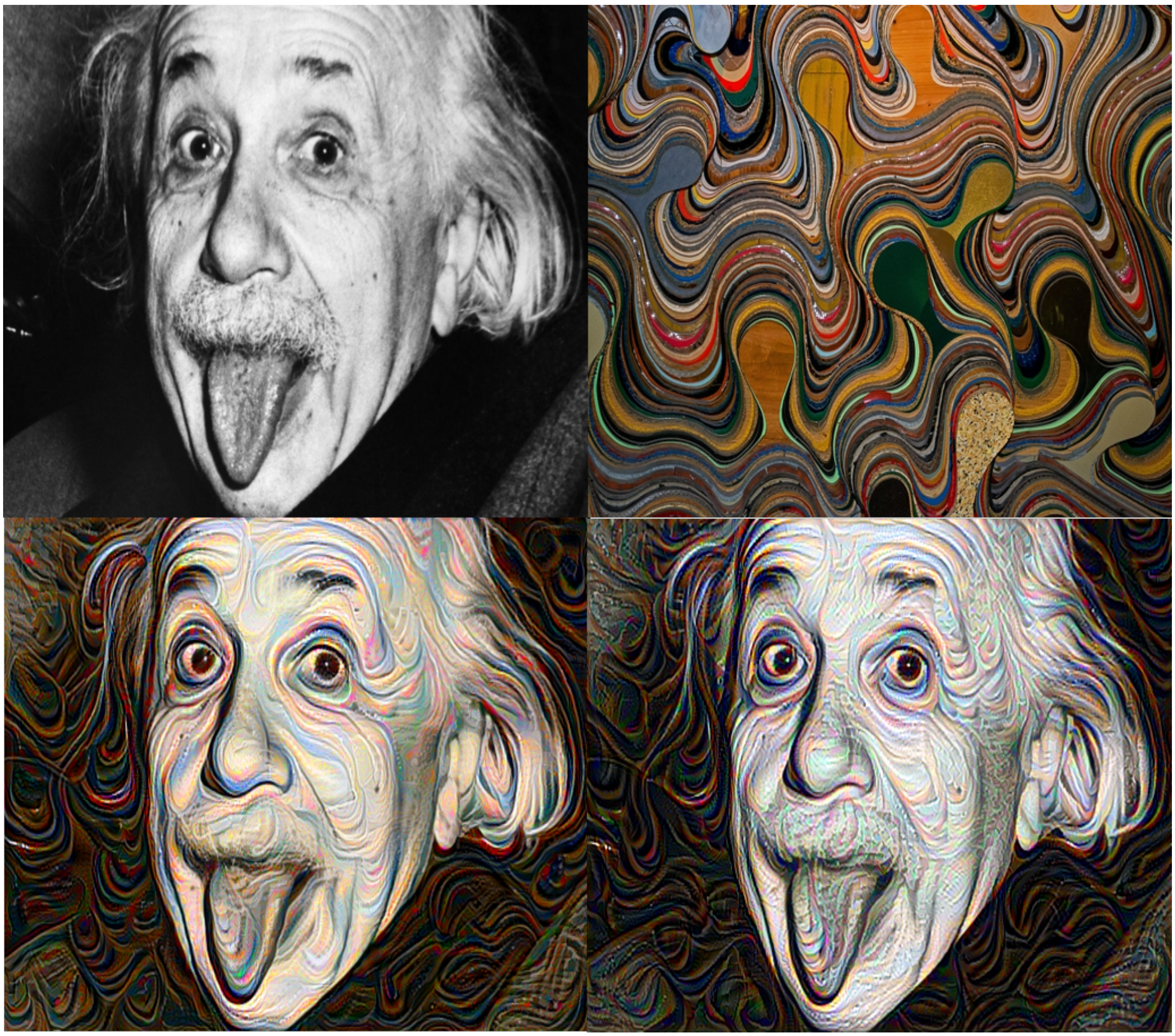
(c) (d)

Figure 3: (a) Content (b) Style (c) NST using VGG16 (d) NST using VGG19 - Adam Optimizer Learning Rate: 0.03

(a) (b)

(c) (d)

Figure 4: (a) Content (b) Style (c) NST using VGG16 (d) NST using VGG19 - Adam Optimizer Learning Rate: 0.01



(a) (b)

(b) (d)

Figure 5: (a) Content (b) Style (c) NST using VGG16 (d) NST using VGG19 - Adam Optimizer Learning Rate: 0.5

(a) (b)

(c) (d)

Figure 6: (a) Content (b) Style (c) NST using VGG16 (d) NST using VGG19 - Adam Optimizer Learning Rate: 5