NEURAL STYLE TRANSFER

20PD05 ANU RAMYA R

## ABSTRACT

Deep neural networks have already surpassed human level performance in tasks such as object recognition and detection. However, deep networks were lagging far behind in tasks like generating artistic artefacts having high perceptual quality until recent times. Creating better quality art using machine learning techniques is imperative for reaching human-like capabilities, as well as opens up a new spectrum of possibilities. And with the advancement of computer hardware as well as the proliferation of deep learning, deep learning is right now being used to create art.

The seminal work of Gatys et al. demonstrated the power of Convolutional Neural Networks (CNNs) in creating artistic imagery by separating and recombining image content and style. This process of using CNNs to render a content image in different styles is referred to as Neural Style Transfer (NST).

NST builds on the key idea that,

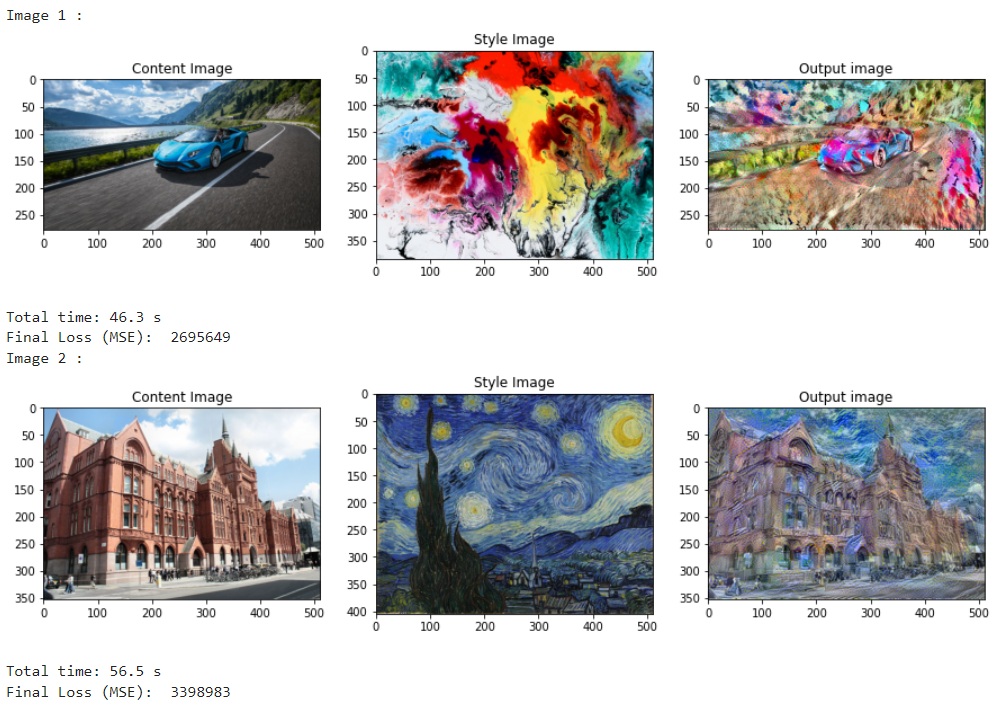
## It is possible to separate the style representation and content representations in a CNN, learnt during a computer vision task (for example, an image recognition task).

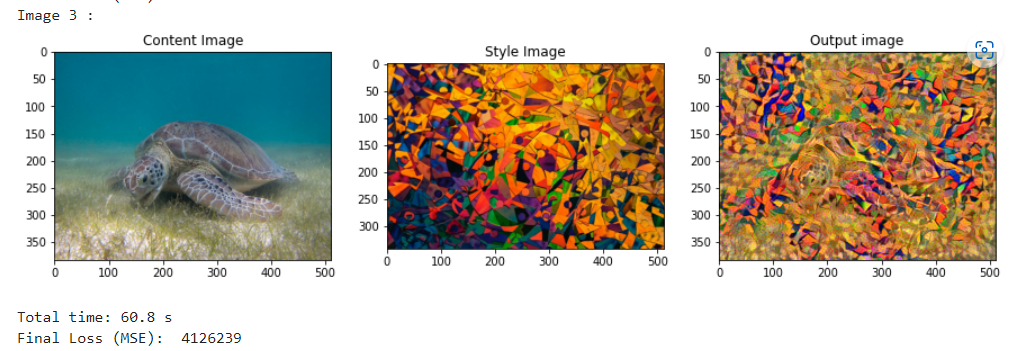
The aim of this project is to implement Neural Style Transfer by using the technique demonstrated by Gatys et al in his paper “A Neural Algorithm of Artistic Style”. A modification of the algorithm is also done to perform Style Transfer with multiple styles.

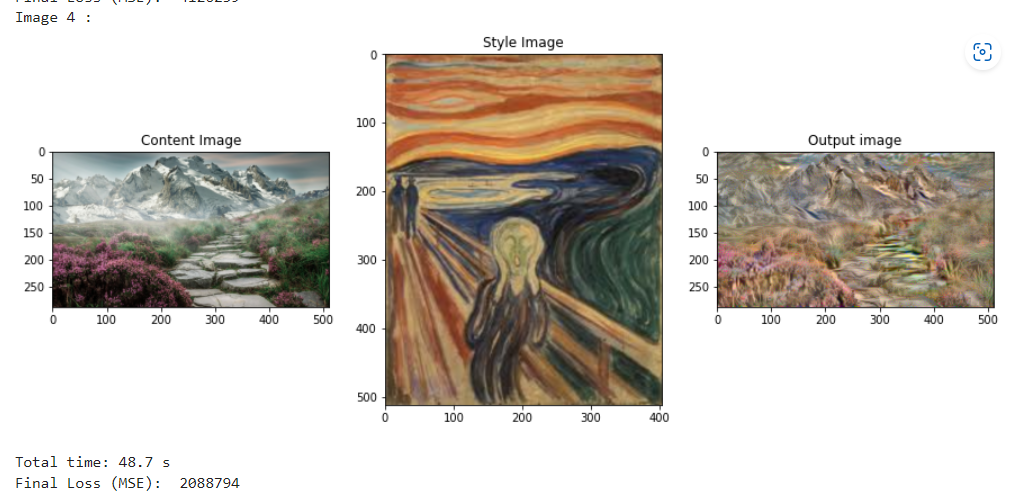
**RESULTS**

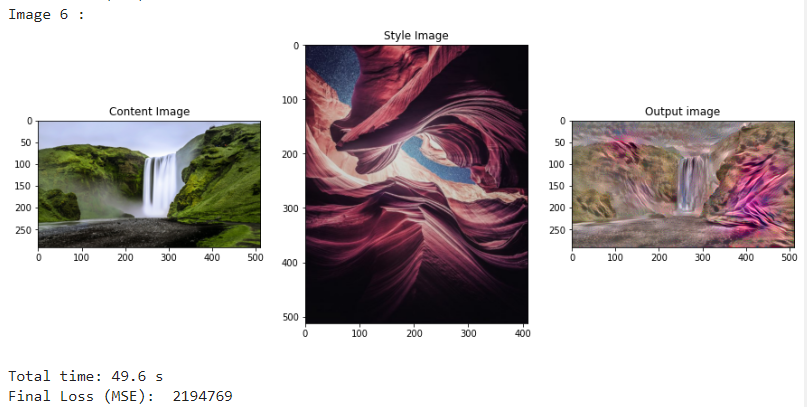
# Testing the algorithm on a set of images to compute average measures:

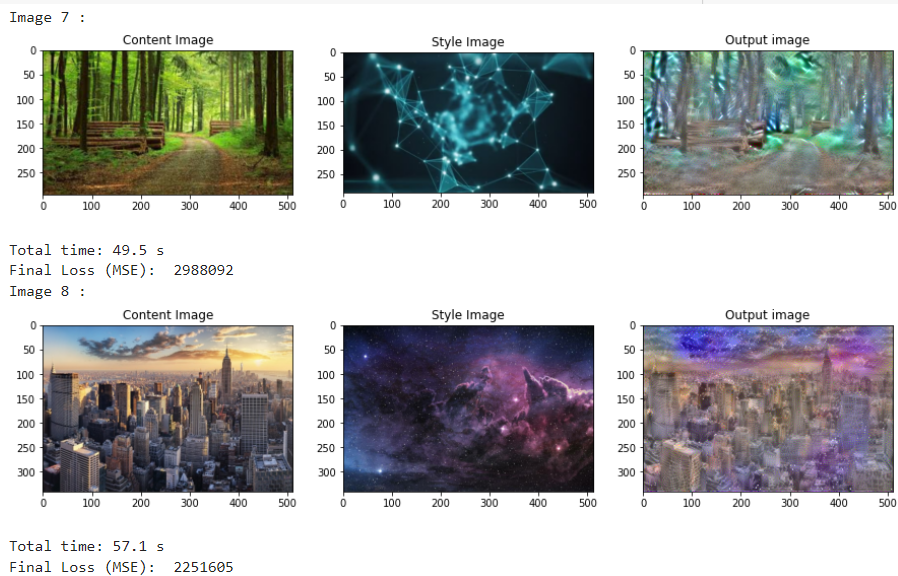
We run the algorithm for 10 sample instances (pair of content and style image) and compute total time and final loss at the end of each run:

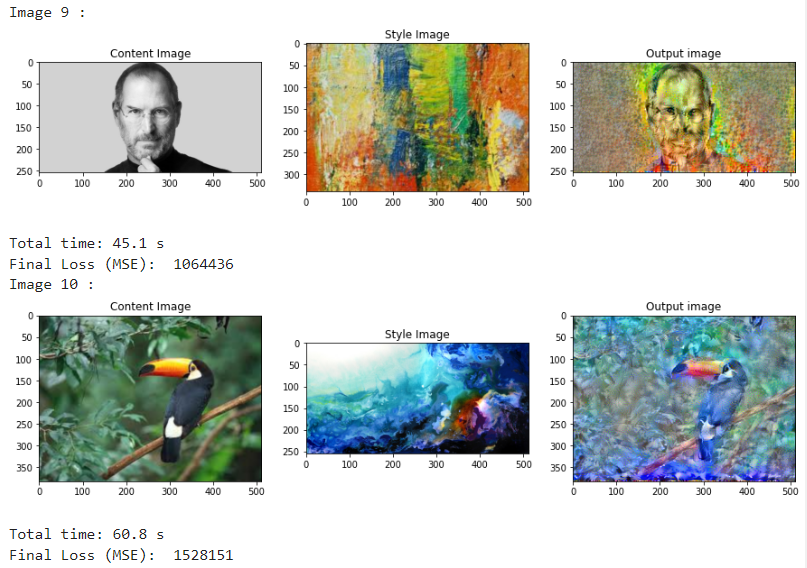












Average time taken to transfer: 53.64 s

Average loss (MSE) after 5 epochs: 2343905

**Modified implementation : NST with two style images**

In the technique discussed earlier, we saw how the generated image extracts content from the content image and style information from the style image.

In this modified technique, we create the output image which extracts content information from a content image, but extracts style information from two different style images.

The architecture of the model remains the same. We use the same intermediate layers to extract content and style. We only make changes to the loss function:

Here, the total loss is given by the weighted sum of content loss, style loss from style image 1 and style loss from style image 2

**L =**  **L content +**  **Lstyle1 +**  **Lstyle2**

Here, α = 10000 , β = 0.1 , γ = 0.1

**Sample run:**

