Neural Style Transfer

An Artistic Application of Convolutional Neural Networks

Presented by Trinity Gahagen

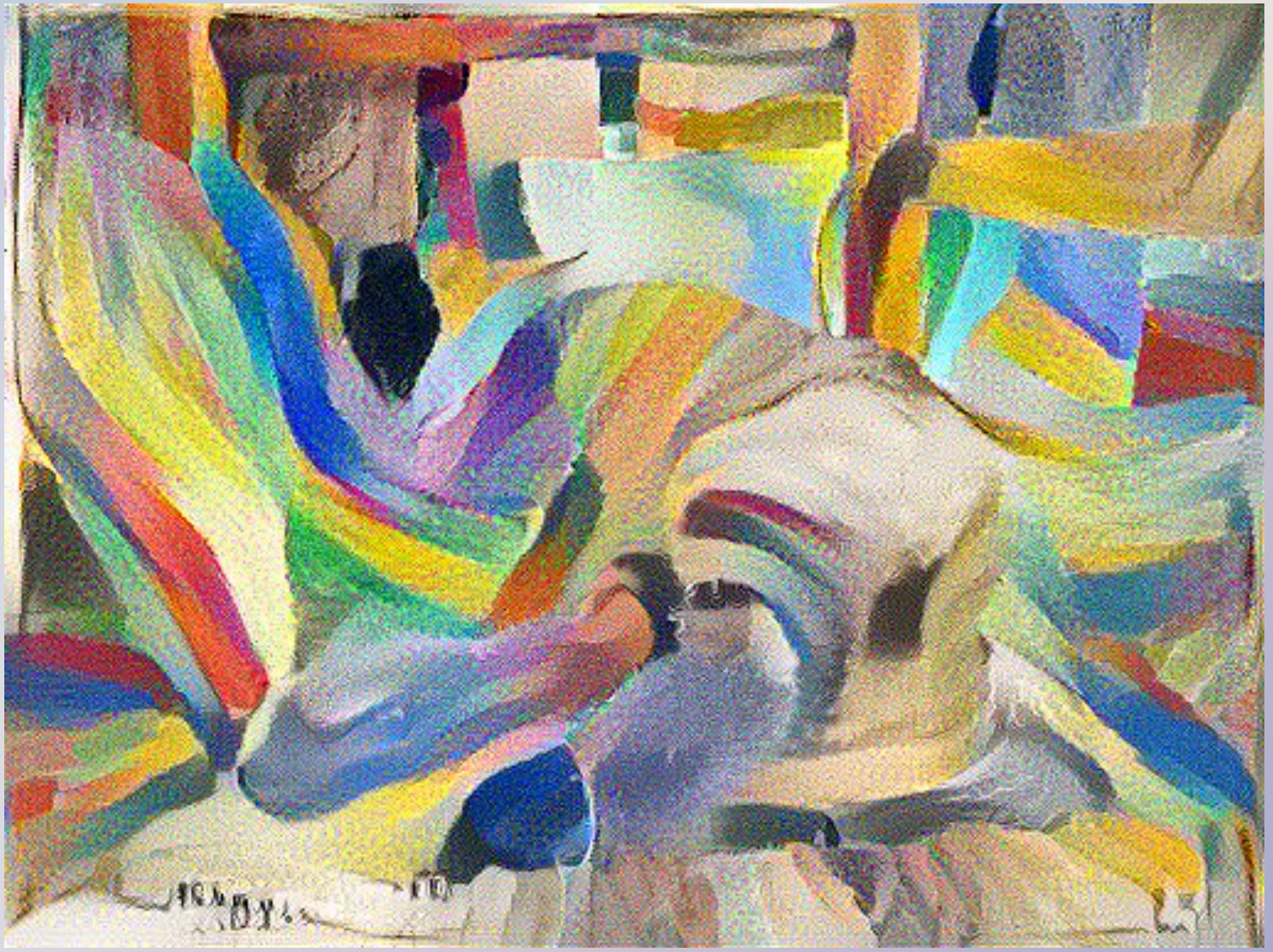
What is it?

Neural Style Transfer is an algorithm that takes the style of an image (like a painting or drawing) and combines it with the content of another image (like a photograph you can take with your smartphone).

For example, it can take an image of this cat…

And combine it with the style of *Political Drama* by Robert Delaunay

To Make This Dramatic Cat



So how does it do that?

The Neural Style Transfer algorithm consists of a few different parts:

1. It extracts features from the content and style images using a pre-trained Convolutional Neural Network (VGG-19), then calculates the “targets”
2. It initializes a third image and extract those features as well, then calculates how *different** this third image is from the content and style images
3. Minimize this difference using some cool math magic

*Difference will be described further in future slides

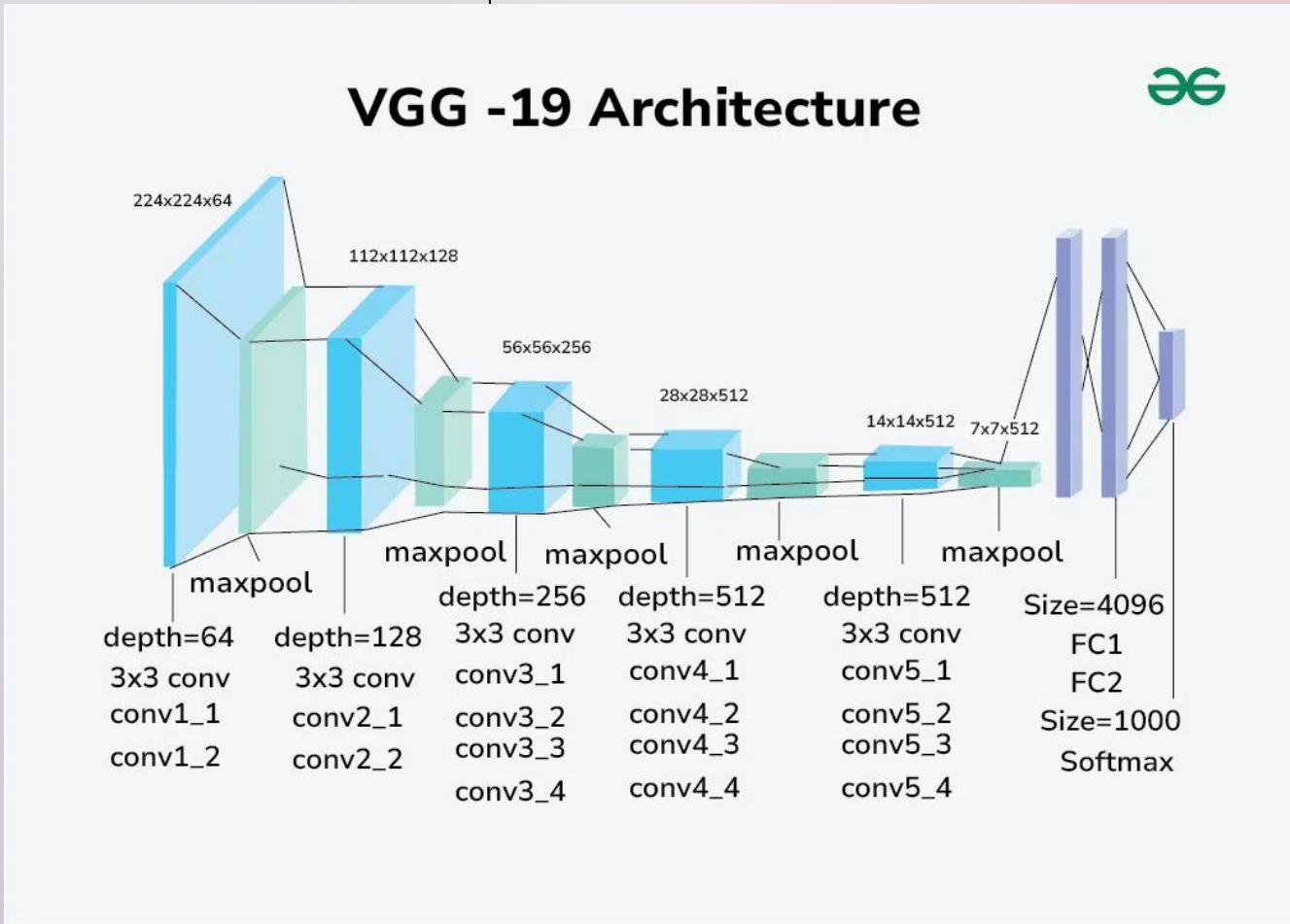
Let’s expand a little.

Convolutional Neural Networks

Capturing the “essence” of a photo

Convolutional Neural Networks (CNNs) can extract the features of a photo as it passes through its architecture and abstracts the photo itself more and more, typically using the features to make a classification.

Neural Style Transfer grabs the different features of a photo that the network extracts at different layers. It does this for the content, the style, and the input image.



How to Measure Difference: *Loss*

$$\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} \left(F_{ij}^{l} - P_{ij}^{l} \right)^2$$

Content Loss

$$\mathcal{L}_{\text{style}}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$$

Style Loss

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

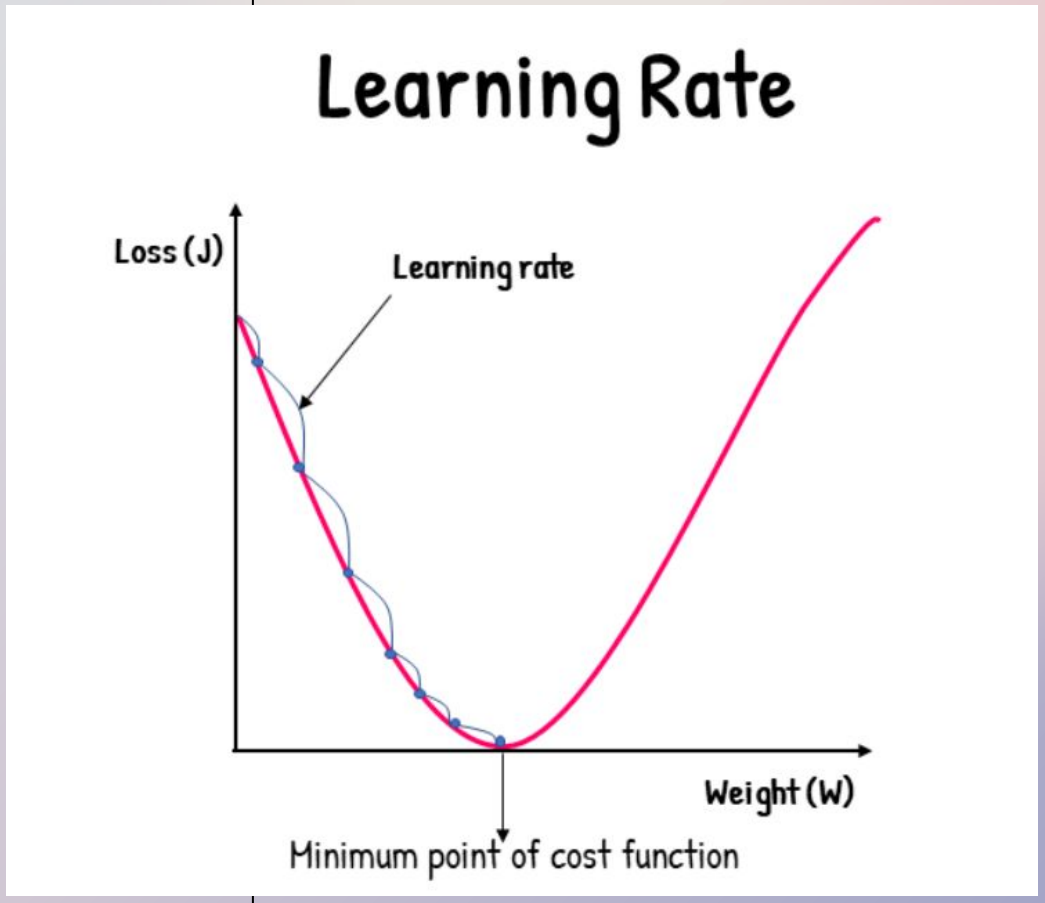
Total Loss

Minimizing Loss Using Gradient Descent

Cool math magic

Derivatives can be used to find the minimum value of a function through a technique called *Gradient Descent*. Gradient descent finds this minimum value by taking steps from an initial guess until it reaches the best value.

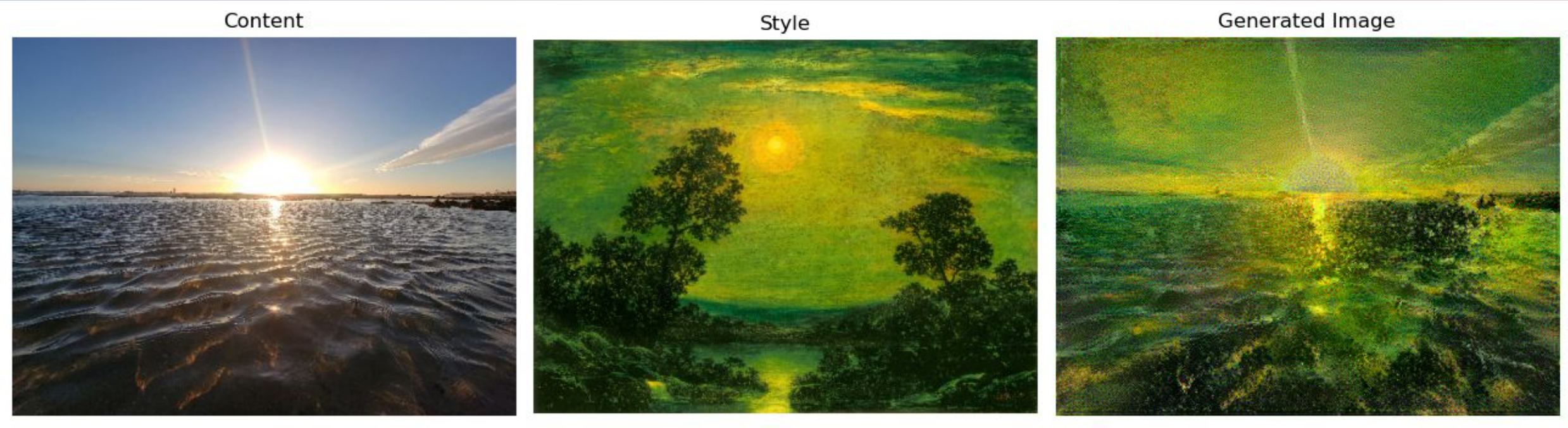
If you'd like to learn more about this specific technique, I would suggest checking out StatQuest's video on it, Gradient Descent, Step-by-Step.



Source: https://editor.analyticsvidhya.com/uploads/28566Slide7.PNG

Once you have all these pieces together…

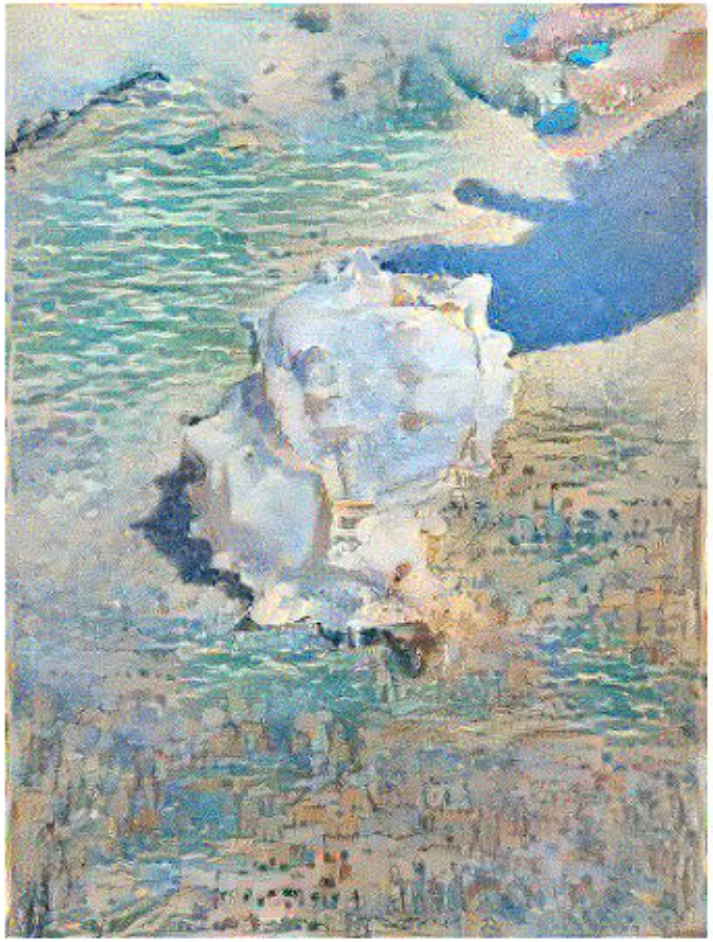
You can create all kinds of images!



Content

Style

Generated Image



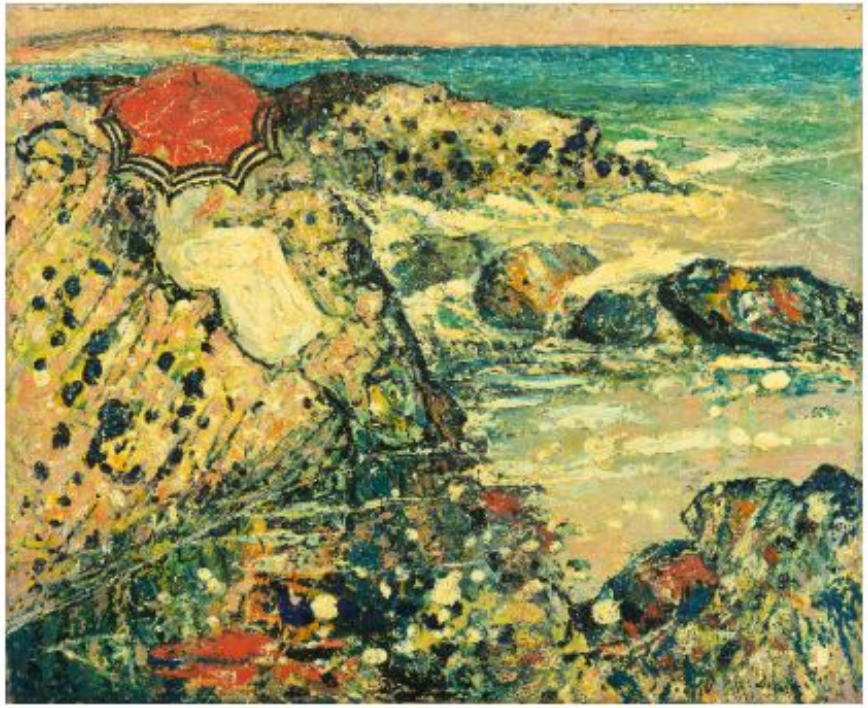
Content

Style

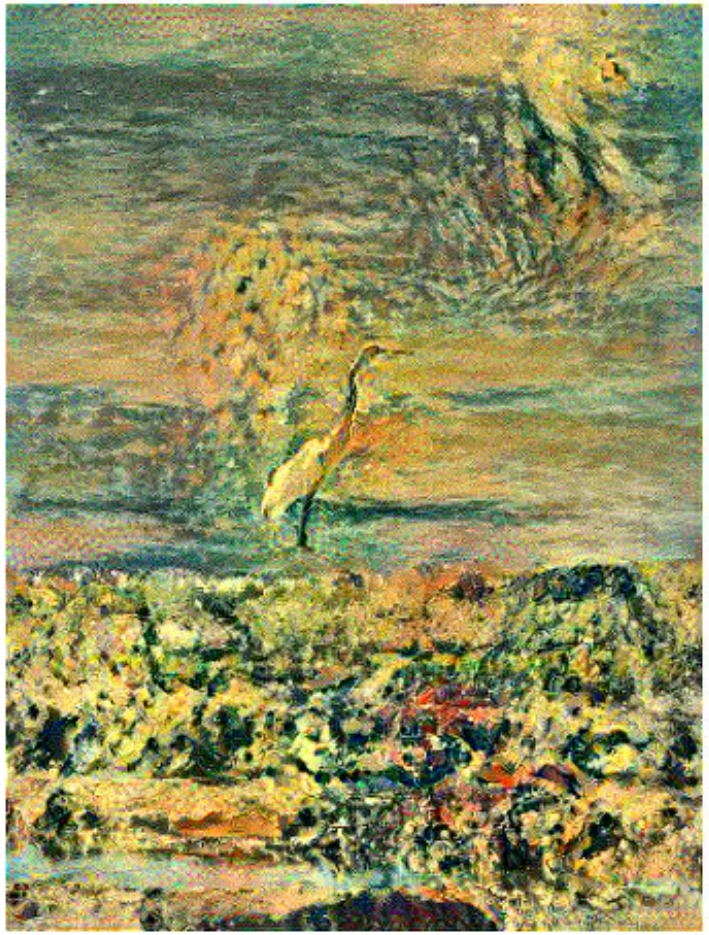
Generated Image

Content

Style



Generated Image



See the Transformation in Action

Thank you for your attendance.

External Resources

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf

https://www.tensorflow.org/tutorials/generative/style_transfer

https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/

Presentation Template: SlidesMania