Style Transfer and Image Analysis

By: Shawn McClain

The Problem

With the A.I. revolution on the horizon, how will Artificial Intelligence transform digital art?

How can neural networks help artists make better art that ultimately changes the world?

Style Transfer

One idea is Style Transfer, which takes a style image and content image, and makes a combination image from the two. The style image provides the style of the image and the content image provides the content of the image.

+

= ?

Clients

Artists

Style Transfer can give artists new tools to be more creative

Businesses

Businesses can transfer art onto products, much like how tie-dye revolutionized the t-shirt

Virtual/ Augmented Reality

Style can be transferred onto games environments or other virtual environments

01

Exploratory Image Analysis

Reading in and Saving Images

```python
#requests, decodes, opens file as array
def image_show(path):
    r = requests.get(path, stream=True)
    r.raw.decode_content
    img = Image.open(r.raw)
    img = np.asarray(img)
    plt.imshow(img)

def image_reader(path):
    r = requests.get(path, stream=True)
    r.raw.decode_content
    img = Image.open(r.raw)
    img = np.asarray(img)
    return img
```

- The Requests library is utilized to grab style images from the internet and save them locally.

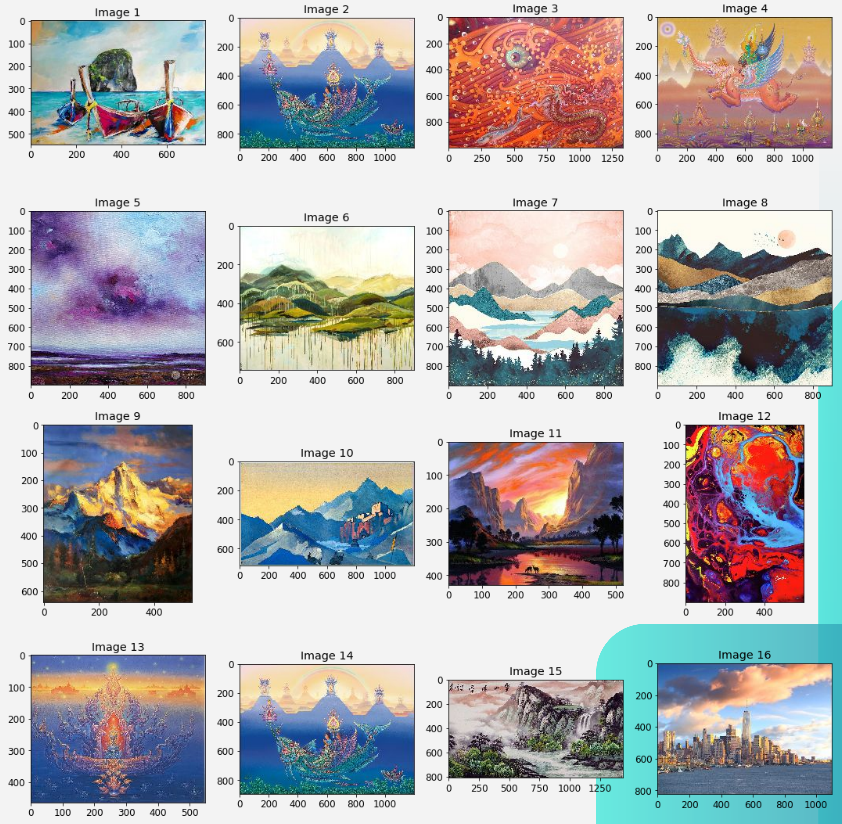
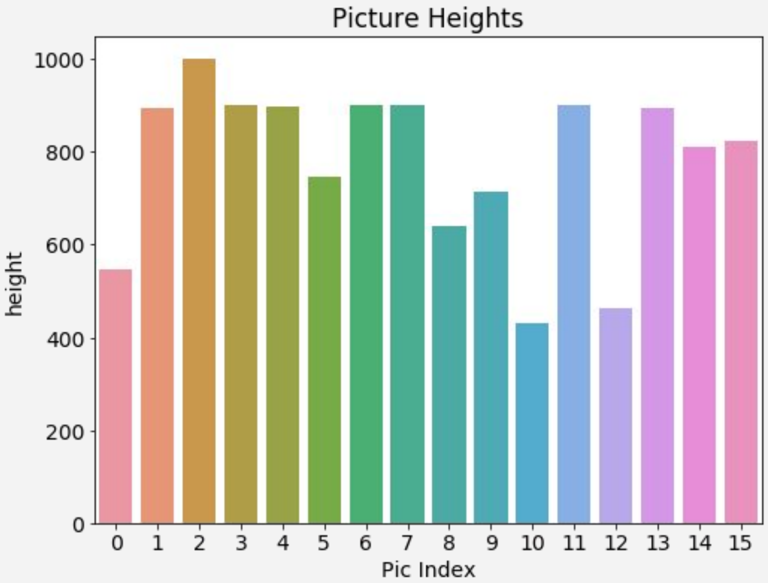
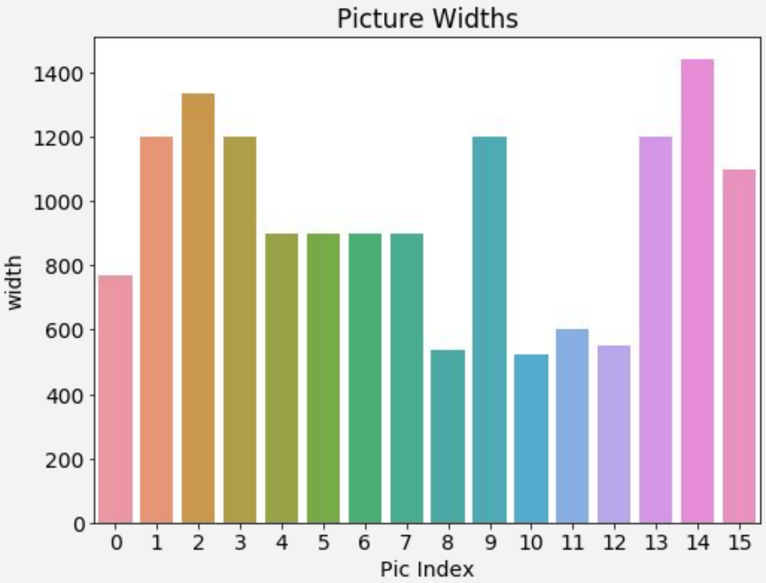


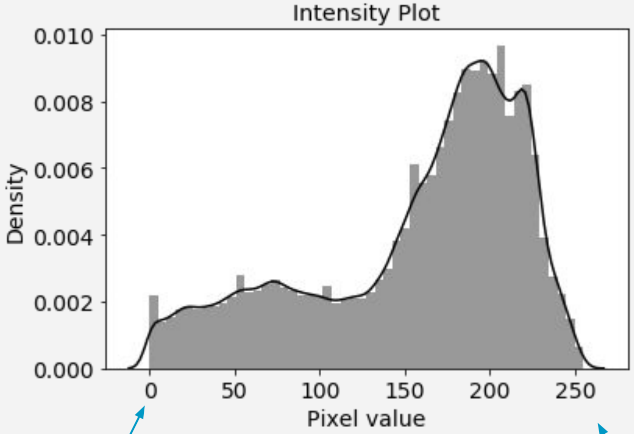
Image Aspects

Our style images tend to have similar aspects that are landscape orientation except for the picture at index 11.





Intensity Plot



Black

White

The intensity plot of the Thai Ocean image shows the density of pixels with light and dark values. **It peaks around 200 which indicates a bright image.**

RGB Intensity

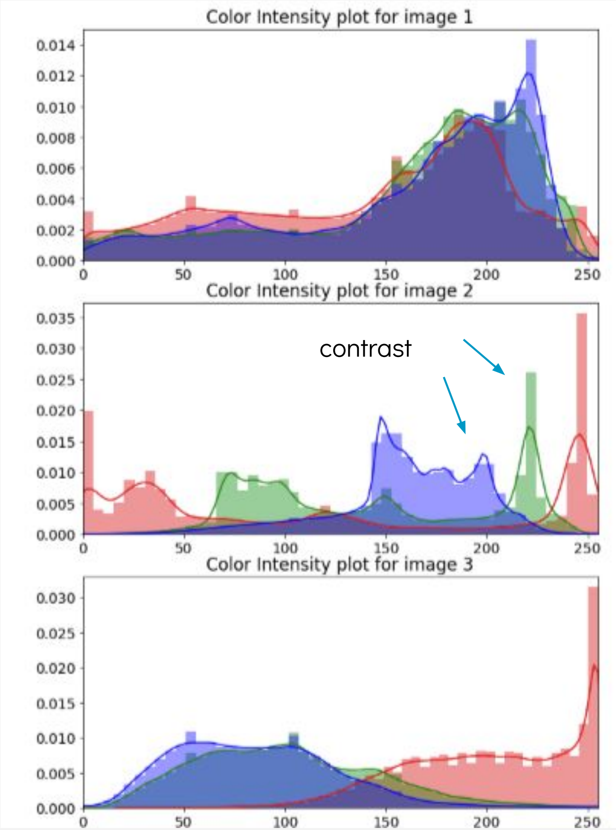


Image 1 has bright blue values that dominant the image.

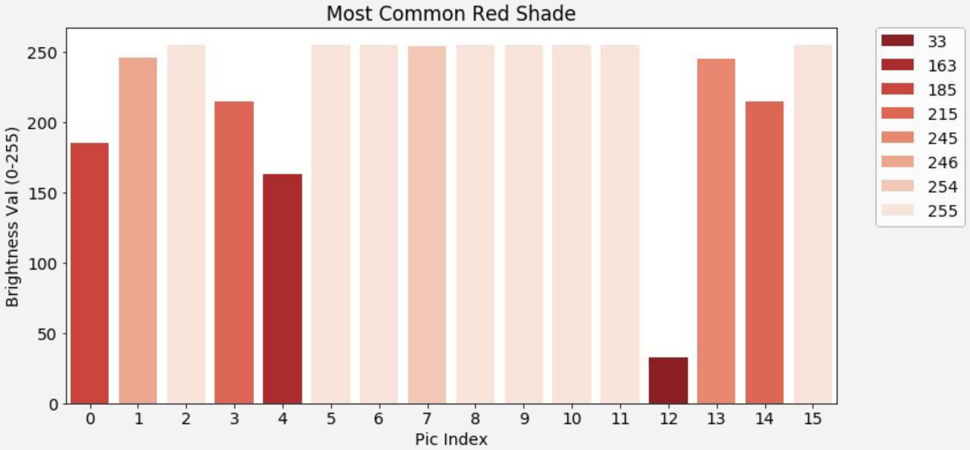
Image 2 has contrast between the 3 channels with bright greens and reds

Image 2 has bright red values that dominate the image and very little greens.

Red Shades

Image
Index 6

Image
Index 12



Finding the mode of the red channel in each image, we find the most common shade of red in each image. (Peak density)

Image 6 has extremely bright reds and image 12 has deep dark reds.

Green Shades

Image
Index 5

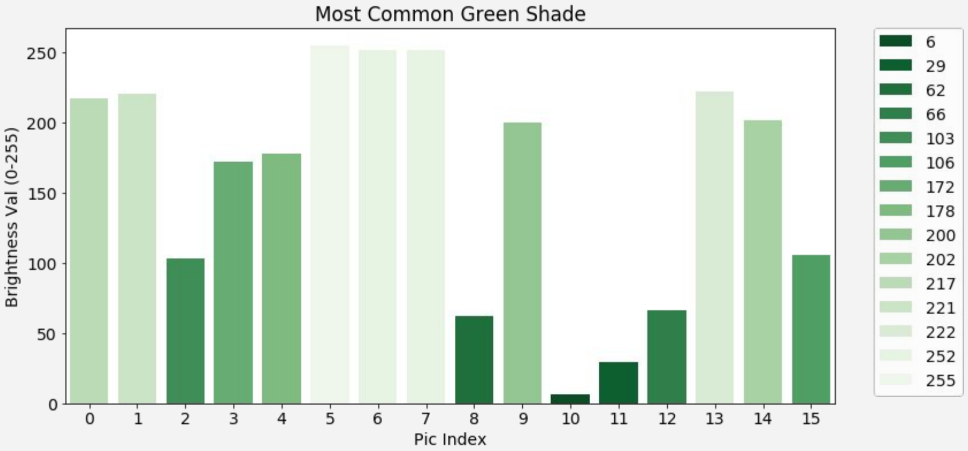


Image
Index 10

Image 5 has extremely bright greens and image 10 has deep dark greens.

Blue Shades

Image
Index 4

Image
Index 10

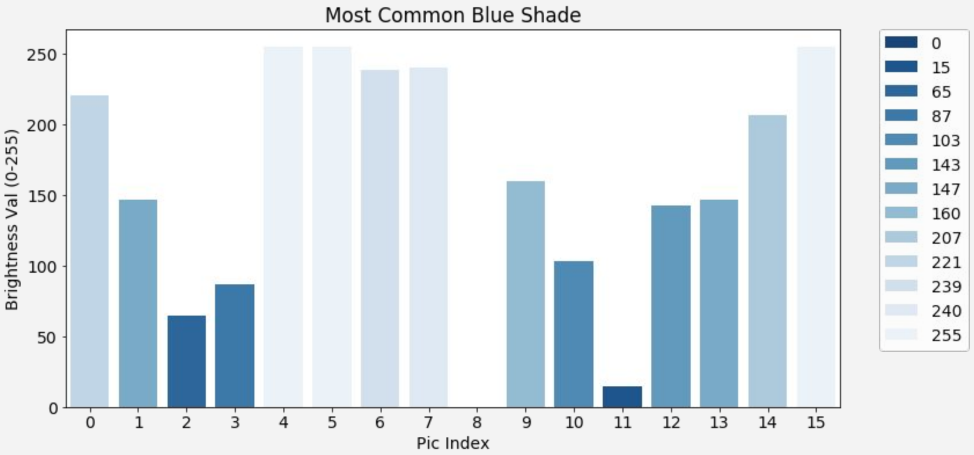
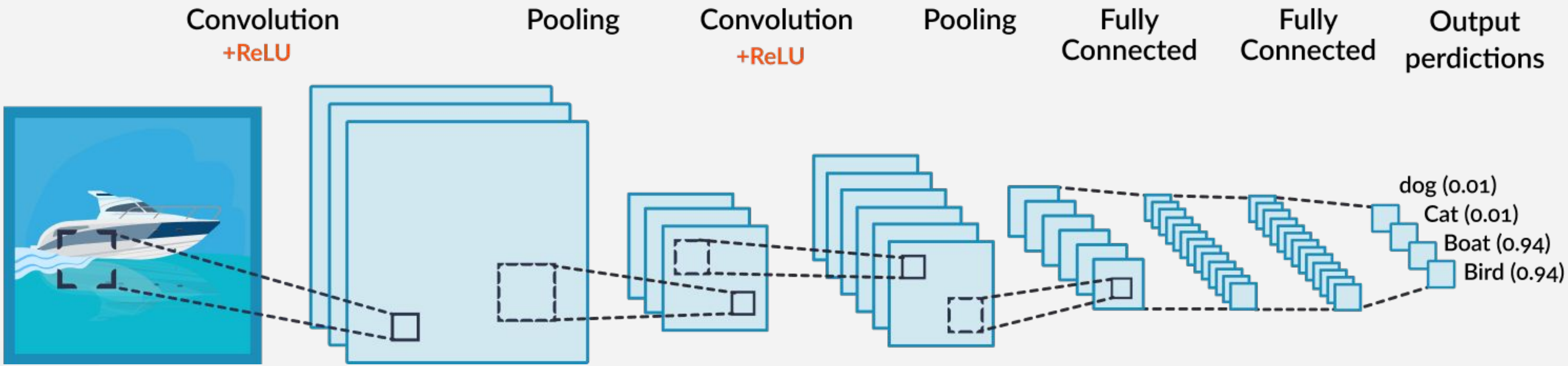


Image 4 has extremely bright blues and image 8 has very dark blue values.

02

Style Transfer Model Explained

Model Architecture



Convolutional neural networks are used for feature extraction. Pixel values are pooled together in multiple layers. The model uses these numeric outputs to signify feature patterns and associate images with those patterns together.

Model Architecture

Style Image

Content Image

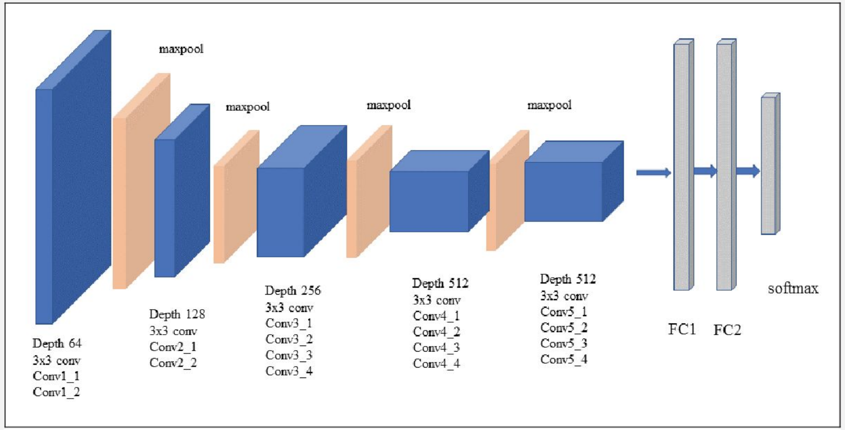


Fig. 3. VGG-19 network architecture

Style and Content photos will be fed into the pre-trained VGG19 model, which is a 5 layer convolutional neural network, that has been trained on the ImageNet database for high accuracy.

Style Image

Model Architecture

Generated Image

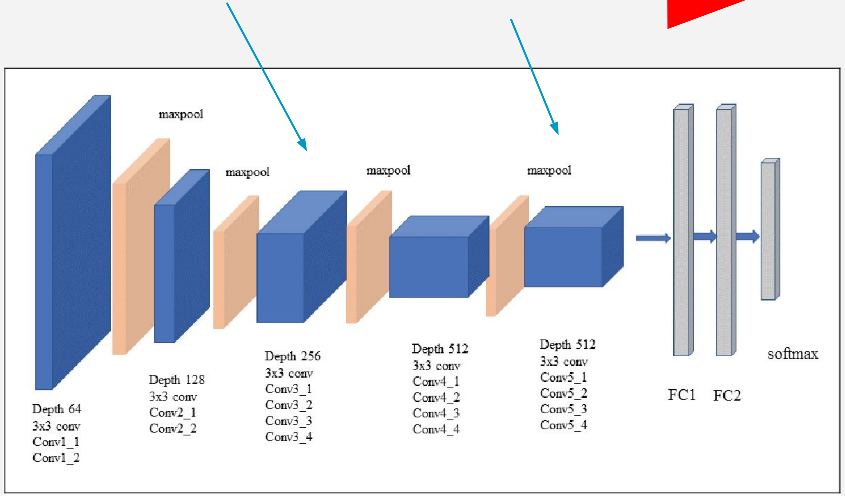


Fig. 3. VGG-19 network architecture

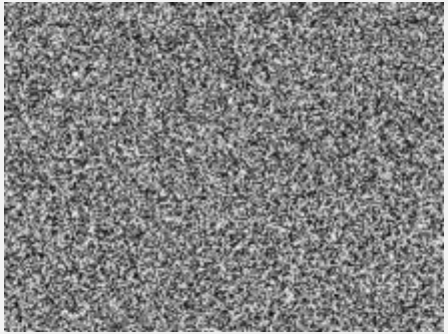
Content Image

At different layers of the model, the loss and gradient are identified for style and content.

Defining Style Loss

Style Image

Generated Image

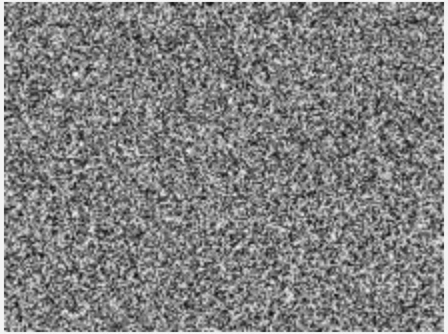


Style loss uses a **gram matrix**, which uses matrix multiplication to represent numerically the style at different levels of the image. This is done by taking the dot product of the flattened pixel values and the transposed flattened pixel values.

Defining Style Loss

Style Image
Gram Matrix

Generated Image
Gram Matrix

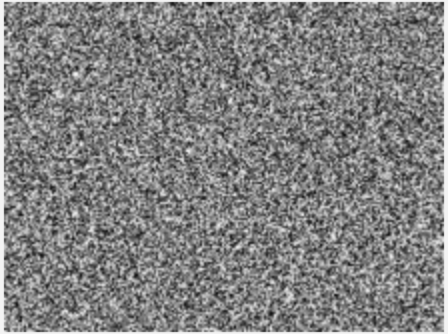
MSE( - )

Once the style and generated gram matrix are defined, the loss is defined as the **mean squared error** between the two, with the style image being our target result.

Defining Content Loss

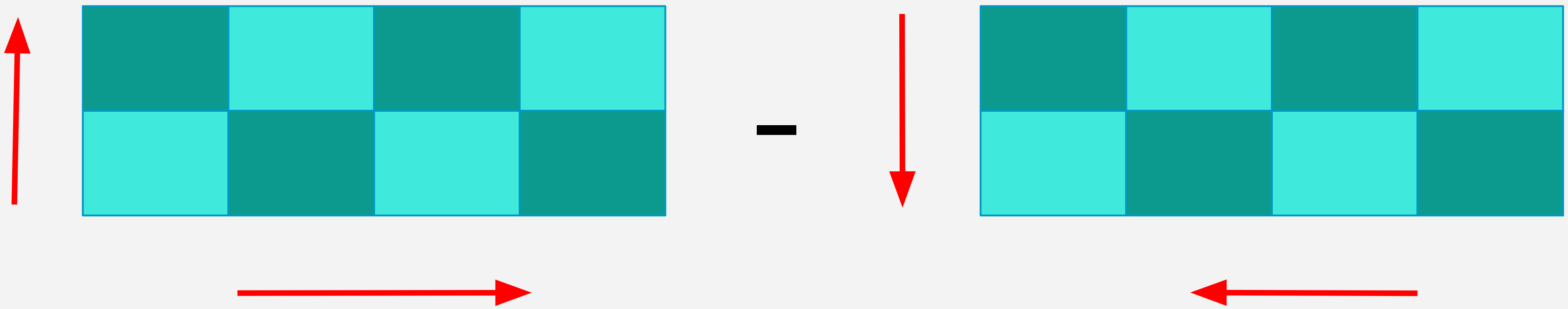
Content Image

Generated Image

MSE( - )

The **content loss** just measures the difference in MSE pixel values between the content and generated images to represent key structures of the content image that we want to target.

Defining Variation Loss



The **variation loss** is an added component that adds **smoothness** to the generated image. This is done by shifting neighboring pixels in both directions and minimizing the difference between them. This makes a gradual transition between borders and content.

Model Architecture

Style Image

Content Image

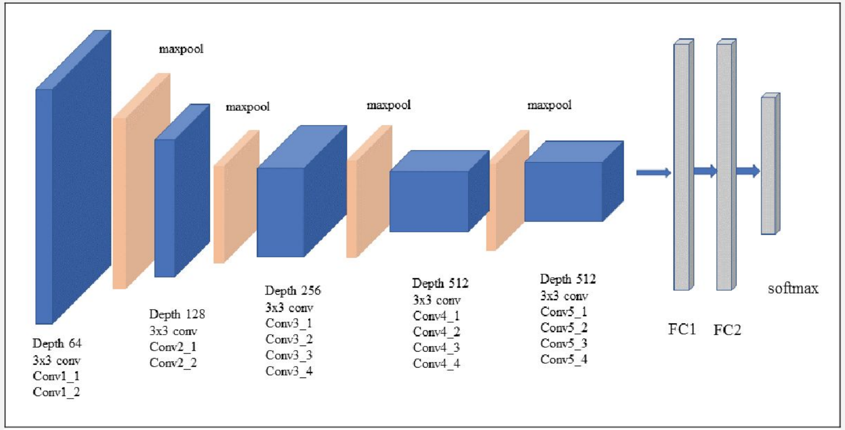
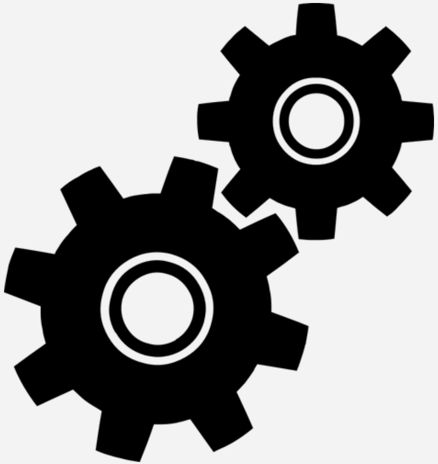


Fig. 3. VGG-19 network architecture

Generated Image

Once the images are passed through, the **L-BFGS algorithm** is used to minimize the loss through **gradient descent**. This process is performed iteratively on the generated image until style loss and content loss are sufficiently minimized.

03



Tuning the Output

Tuning the Output

In many machine learning models, we could use a grid search over parameters to maximize accuracy or precision of the model. But more iterations (minimized loss), does not guarantee a more aesthetically pleasing image. Therefore, I did a manual grid search through the weights to find the best result

Params to tune:

1. Style Weight
2. Content Weight
3. Variation Weight
4. Number of Iterations
5. Selection of Style or content image

Tuning Content Weight

Standard Starting Point:

Number of Iterations- 15
Content Weight - **[.01,.025,1, 10]**
Style Weight- 1
Variation Weight -1

Content Image

Style Image

Content Weight Results

Content Weight: .01

Content Weight: .025

Adjusting content weight didn't do much to change the image except darker colors appeared slightly lighter with a higher content weight. **I think 0.025 is fine as the standard choice.**

Content Weight: 1

Content Weight:10

Tuning Style Weight

Standard Starting Point:

Number of Iterations- 15
Content Weight - **.025**
Style Weight- **[.01, .1, 1, 10]**
Variation Weight -1

Content Image

Style Image

Style Weight Results

Style Weight: .01

Style Weight: .1

Style Weight: 1

Style Weight:10

Adjusting style weight had an arbitrary difference on the output. The style weight of 0.01 might be slightly darker. But otherwise, I think the weight can remain at 1.

Tuning Variation Weight

Standard Starting Point:

Number of Iterations- 15
Content Weight - **.025**
Style Weight- **1**
Variation Weight -**[.5, 1, 2, 3]**

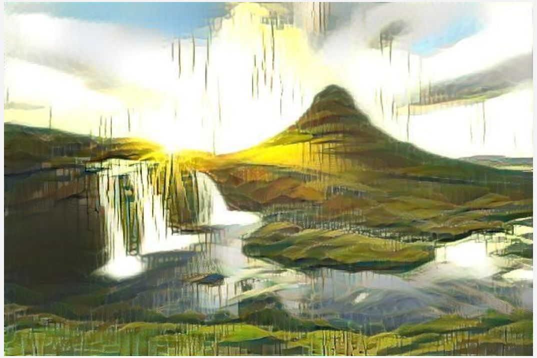
Content Image

Style Image

Variation Weight Results

Variation Weight: .5 (did not run)

Variation Weight: 1



A higher Variation Weight actually created more lines from the dripping in the style image. A variation of .5 produced a null loss value. I think the value of 1 looks the best.

Variation Weight: 2

Variation Weight:3

Tuning Number of Iterations

Standard Starting Point:

Number of Iterations- **[5, 15, 50, 100, 200]**
Content Weight - **.025**
Style Weight- **1**
Variation Weight -**1**

Content Image

Style Image

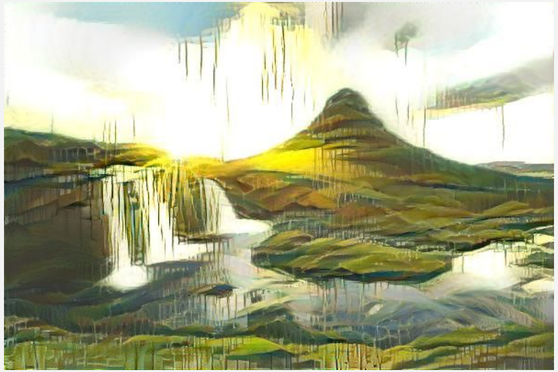
Iteration Results

Iteration 5

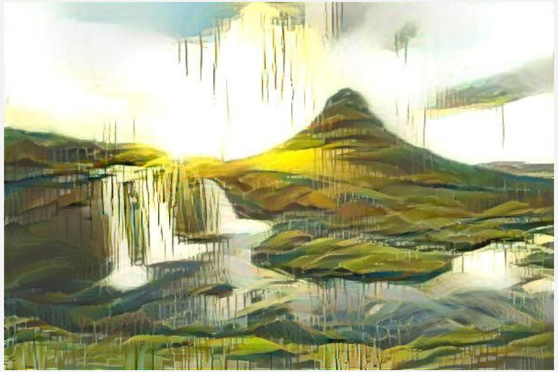
Iteration 15

More iterations had a noticeable difference on the results. More iterations exposed more of the shadows of the content image. The sky became whiter with more iterations with less clouds. I settled at 15 iterations for a more natural look.

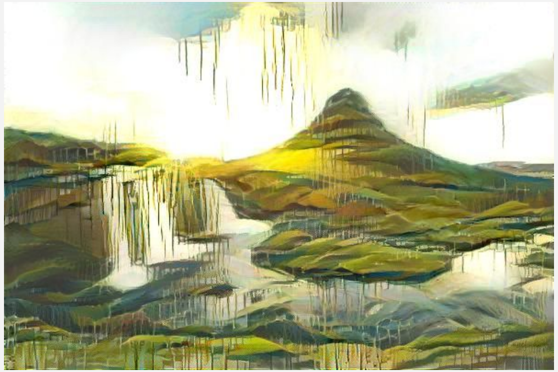
Iteration 50



Iteration 100



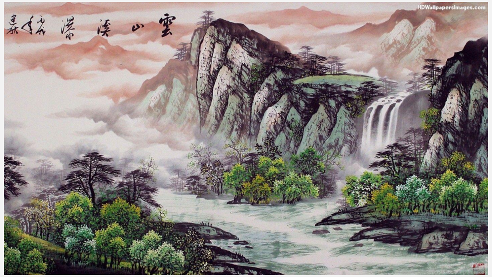
Iteration 200



Weights:
Number of Iterations- **15**
Content Weight - **.025**
Style Weight- **1**
Variation Weight -**1**

Experimenting with Other Images

Experimenting with other style pictures, most have decent results, but pictures that have less details and variation with style seem to look the best.



In Conclusion

Style Transfer with the VGG model provides good results to combine the style of one picture with the content of another picture. Through the use of a gram matrix, we can interpret style and its loss. But there is still much to be desired that could make style transfer that much better. Perhaps we don't want all of the image to have style transferred over and some parts of an image (like its shadows) left as is.

As far as tuning the image, more iterations and higher weights don't seem to make an easy fix to improve the quality of the result. A lower loss does not guarantee a more aesthetic result. Perhaps there are more losses that can be added to future models to improve results even further.