Project #2: Neural Style Transfer – Proof of concept

Michael Ersevim

Data Science, Bellevue University

DSC 680: Capstone Course

Dr. Brett Werner

Spring, 2023

Github Repo: https://github.com/Actuweary/Projects.git

(This Project will be stored in folder named 'Proj 9')

Project Outline:

The ultimate purpose of this project is to develop Python code which will learn the style of a painting and apply it to another image creating a pastiche of the two images.

Background:

Artificial intelligence (AI) has revolutionized the way we create and consume art. With the help of machine learning algorithms, it is now possible for computers to learn an artist's style and generate new works of art in that style. This technology has given rise to a new form of art known as AI-generated art, which has gained significant popularity in recent years. However, as with any new technology, AI-generated art raises several ethical and moral questions. One of the primary issues surrounding AI-generated art is the question of authorship. Who should be considered the creator of the artwork—the artist who trained the machine learning algorithm or the algorithm itself? This essay will explore this question and other issues surrounding AI-generated art, particularly in relation to an artist's style and prior work.

One of the most significant challenges with AI-generated art is that it blurs the line between the creator and the tool used to create the artwork. In the traditional sense, an artist's style is unique to them, and it is a product of their individual experiences, skills, and creative decisions. However, with AI-generated art, the artist's style is learned by the algorithm, and it is the algorithm that creates the artwork. This raises the question of whether the algorithm should be considered the artist or merely a tool used by the artist. On the one hand, some argue that the algorithm is the creator of the artwork, and therefore, it should be given credit as such. After all, the algorithm is responsible for the creative decisions that went into making the artwork, and without the algorithm, the artwork would not exist. However, others argue that the algorithm is merely a tool used by the artist and that it is the artist who should be credited as the creator of the artwork.

Another issue with AI-generated art is the question of originality. While the algorithm may be able to learn an artist's style and create new works of art in that style, it is not capable of producing truly original artwork. This is because the algorithm is limited by the data it is trained on and is unable to think creatively or make unique artistic decisions. As a result, AI-generated art may be viewed as a copy or imitation of the original artist's work, rather than an original creation. Moreover, there is the risk of AI-generated art becoming formulaic, as the algorithm relies on the patterns and structures it learns from the artist's work. This could result in a lack of diversity and creativity in the artwork produced by the algorithm, which could ultimately lead to a loss of interest in AI-generated art as a whole.

Another ethical issue with AI-generated art is the question of ownership. Since the algorithm is responsible for creating the artwork, it is unclear who should own the rights to it. Should it be the artist who trained the algorithm, or should it be the developer of the algorithm? If the latter is true, then there is the risk of developers owning the rights to all AI-generated artwork, effectively monopolizing the industry. Furthermore, the use of AI-generated art raises questions about the role of the artist in society. If AI-generated art becomes more prevalent, will it reduce the value and significance of human creativity and artistic expression? Additionally, there is the concern that AI-generated art could replace human artists, leading to job loss and a decline in the importance of art as a human endeavor.

AI-generated art raises several ethical and moral questions, particularly in relation to an artist's style and prior work. The question of authorship is a significant challenge, as the algorithm blurs the line between the creator and the tool used to create the artwork. Additionally, there is the question of originality, as the algorithm is not capable of producing truly original artwork. Moreover, the issue of ownership is unclear, as it is uncertain who should own the rights to the resultant works. Undoubtedly, there will be many a court case which may help draw the lines in this developing battlefield.

Broad steps to be applied:

1) Import appropriate Python libraries
    a. Various common: MatPlotLib, Tensorflow, Numpy, OS
    b. Less common: PIL, Colab, Keras.processing
2) Import and pre-process images
    a. Base image, Style image
3) Import pre-trained model
    a. VGG19
4) Create new image with style applied and save it
5) Allow input/output files with adaptability to varying styles and resolutions

Likely questions from audience:

1) How long did it take to train the model?
    a. The model is actually pre-trained which can be called in to do the Neural Style Transfer very quickly. It only seconds, even with a CPU.

2) What resolution of pictures did you use for Base and Style pics?
    a. The Base image was much higher quality with 2016 by 1512 pixels and a 2MB file size while the Style Image was 240 by 318 pixels and a file size of only 175kb

3) How do you import the file locally instead of a path?
    a. This was probably the trickiest part of the project. There are several examples where the images are a path to a location on a web-site. This presented some file type issues with the processing. Using the OS library, I could point to the local directory.

4) How many different pre-trained models are there for applying styles?
    a. This project leveraged the VGG19 model, but there are many models and a table of the most popular models is listed in the ***appendix***

5) What ethical considerations did you consider?

   a. I spoke at length with the artist and she was very gracious and helpful with providing dozens of images, any of which could be used to serve as the ‘style’ image, from which to train. She was mostly agnostic to the techniques and purpose, other than experimentation and proof of concept. She did wonder aloud if or how she could leverage and apply some creation/production techniques in the future. She knows that I would not utilize her painting images for anything further than the project I described without her consent.

6) Can anyone use this tool/technique?
   a. Yes, since it is proven to work in Colab, anyone could create an account and run it.

7) Is a GPU needed to run this code efficiently?
   a. No – the code ran very quickly on a Colab CPU, only taking seconds to run the model and produce the pastiche output.

8) What were the three respective files sizes used/created?
   a. See above for the inputs. The output file was 1.1MB with a pixel size of 1,024 by 768.

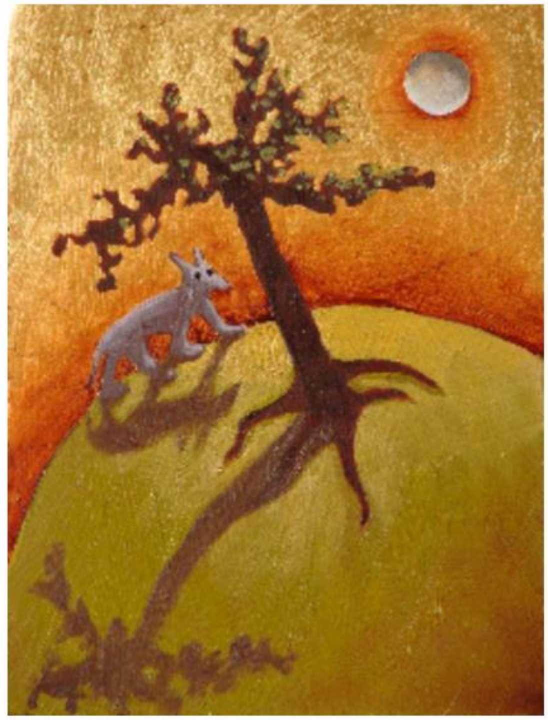
9) What did the artist think of the created pastiche?
   a. She was ‘amazed and a little scared’. She also commented that the output looked as is some sort of ‘paint’ filter had been applied, a setting I assume is found in commonly used graphic packages.

10) What other AI generated art projects will you investigate next, if any?
   a. I will experiment in the future with more examples where the computer generates an entirely new image from a random ‘seed’ after learning from many examples of an artist’s corpus of works.

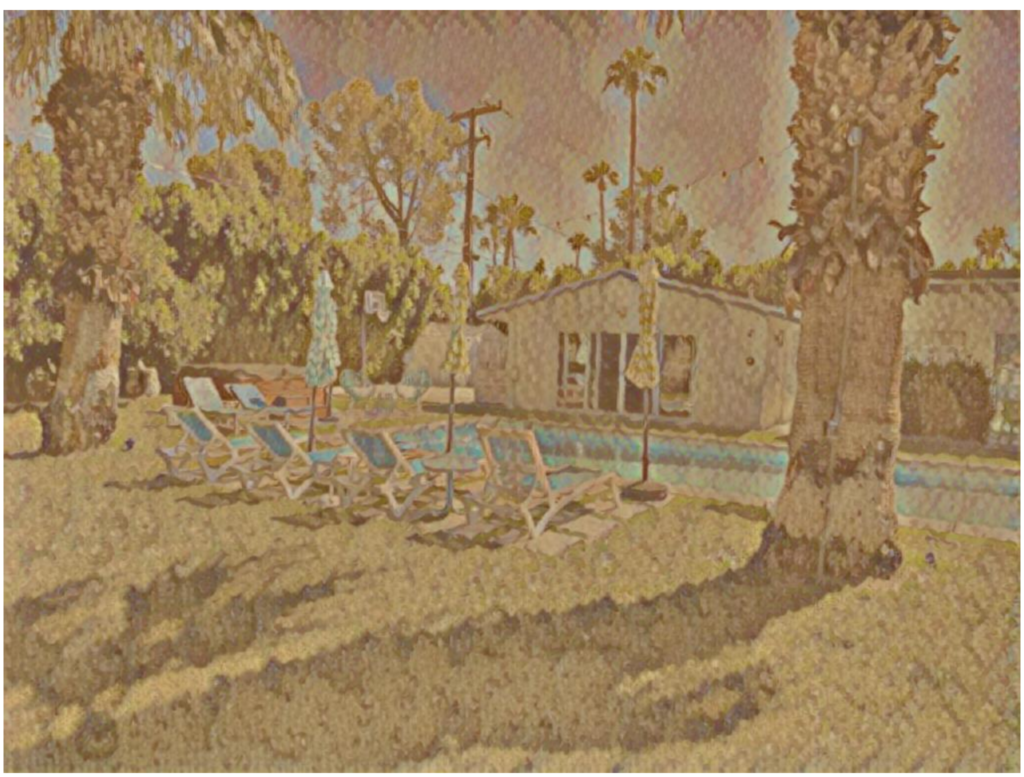
Illustrations: *('Style Image', used with permission on the artist, Christine Brennan)*

Content Image

Style Image



<u>The resultant stylized pastiche image:</u>



Appendix

List of most popular computer-vision pre-trained models:

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.79	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.9	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.76	0.93	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.78	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.75	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.96	88,949,818	-
EfficientNetB0	29 MB	0.771	0.933	5,330,571	-
EfficientNetB1	31 MB	0.791	0.944	7,856,239	-
EfficientNetB2	36 MB	0.801	0.949	9,177,569	-
EfficientNetB3	48 MB	0.816	0.957	12,320,535	-
EfficientNetB4	75 MB	0.829	0.964	19,466,823	-
EfficientNetB5	118 MB	0.836	0.967	30,562,527	-
EfficientNetB6	166 MB	0.84	0.968	43,265,143	-
EfficientNetB7	256 MB	0.843	0.97	66,658,687	-