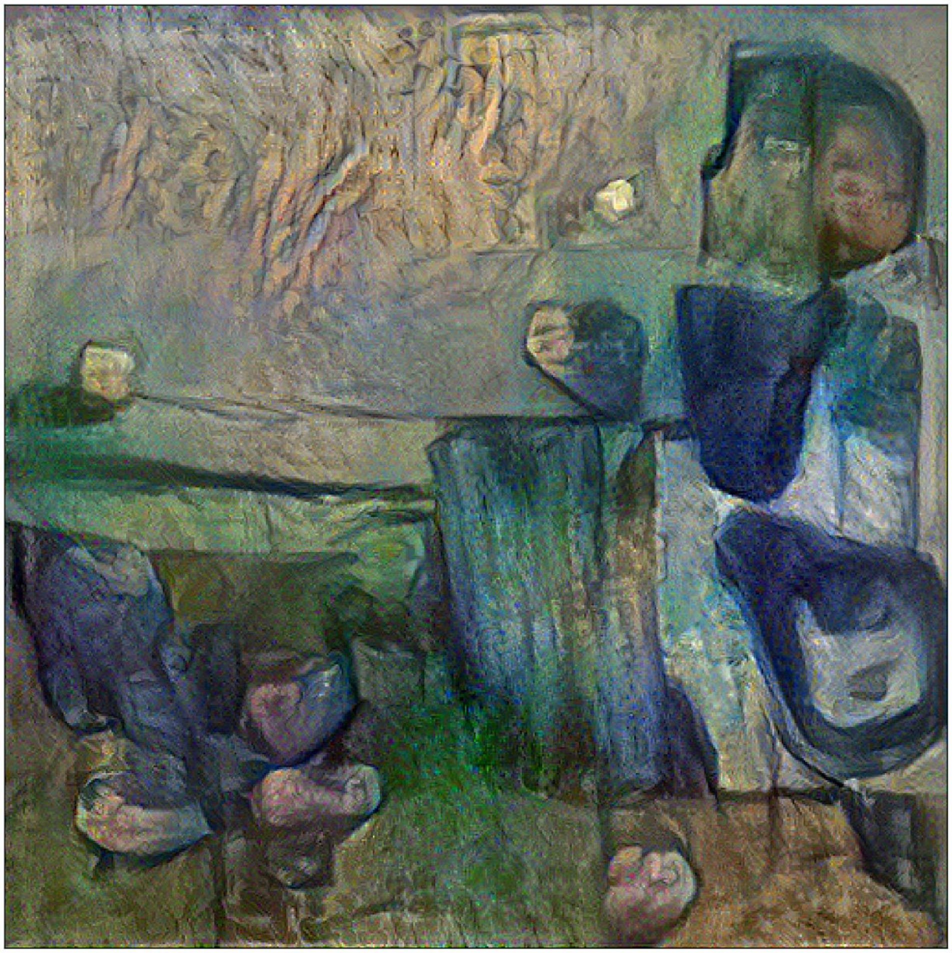
Machine Learning for the Arts
UCSD SPRING 2019
FINAL PROJECT

Artist Generation



Painting by Baltasa Stinstoken, just remockation. - Rudyard Kilker

Insist listen for most of laughter.

Brian Henriquez

DESCRIPTION

Given that I had the most luck when generating quotes from the r/Quotes subreddit, I thought I could retouch this idea but in a different approach. Originally, the intention was to mimic a user and simply generate quotes that other users may find interesting. However, given how we have already learned how to generate images, I felt that this project could use some visual aids. These visual aids would make more sense if we were to consider the people who uttered the quote instead. That is, instead of being interested in the user space, we are now more interested in the author space and what these quotes can tell us about them. Seeing that networks can generate images from text, I felt that a perfect use for such a network would be to generate images based on generated inspirational quotes. These images would then serve as a visual parallel to the quotes.

The style transfer portion of this project was actually part of the motivating factor. While AttnGAN does indeed generate images from any given input text (that has some embedding), it does not generate images that are stylistically interesting (unless trained on such images). Generating a regular dataset large enough to satisfy training the network was out of the question, so instead a neural style transfer algorithm was used in order to make the results more visually appealing. The finalized project was a success as the generated images are indeed interesting to observe. The outputs have been provided in this repository and will be presented in the form of a presentation in the delivery date.

Note that while a lot of these networks were covered in class, what I really sought was the novelty in the way that these networks were chained together. This pipeline can essentially create hundreds of "artists" and their respective artworks and a quote describing them. Instead of simply generating artwork, this network first generates the person and then the artwork. This is what makes this pipeline more interesting than most others.

CONCEPT

Originally, I just wanted to do something with the quotes that were generated from P1 due to how interesting and moderately varied they were. I really wanted to create some visual aspect of it in order to make it more interesting, but it didn't make sense to generate images from names as they hold no meaning. However, since speech does give some insight into the mindset of others, creating visuals based on the quotes would make much more sense!

So, I came up with the idea that based on these quotes, different types of artworks would be generated. In essence, based on the opinions of many, a network would be "creating" an artist that has a particular saying. Artwork that would be created from this saying would effectively also be a parallel to the quote and very much belong to the fictional person in question.

I am aware that calling this pipeline an "artist generator" is bound to get some bad looks mostly because there are many people that believe that machines cannot be the artist. But in this pipeline of creating the person and then the artwork, does it really makes sense to believe that the artwork is actually the work of the algorithm designer? Not really. The original sources are technically the thousands of people used for the original seeding of the quote generator. So, while I may be the one to have loaded all of the data, perhaps the sources of the data have more directly taken a part in the pipeline.

TECHNIQUE

This project, by part, is quite simple, but it's the fact that several networks have been placed together that makes it hard to achieve in a single network. To start, the same char-RNN from the first project was used. The RNN used consisted of LSTM units trained for 100 epochs on a dataset of slightly over 30k quotes collected from the r/Quotes subreddit. Training converged fairly quickly, and improvements after the 100th epoch or so tended to increase epoch training time by a factor of 10, so I had to settle for 100 epochs (initially I had used 300). This network was trained to try and generate quotes of the following form:

"[Quote Text]" - Author

The second part of the network is a word-RNN that was trained on a much smaller corpus for sentiment analysis for about 20 epochs. Since the encoder itself only knew words that the network was initially trained on, words that were unknown in the quote text used during evaluation (which was frequently an issue due to the way the char-RNN works) were simply evaluated as extra padding. What that meant was that the word itself carried no intrinsic meaning in the sentence.

Finally, with the quote and sentiment extracted from the quote in hand, we now generate the visual based on the input quote. The network that generated the images was AttnGAN. This network is effectively an image encoder-decoder that was then used in the training of a text encoder to image decoder setup. I used a network that had both of its components pre-trained on the COCO dataset. Since the text encoder wasn't simply a dictionary lookup, no extra care needed to be taken for words not seen before. This network generated the "base" image.

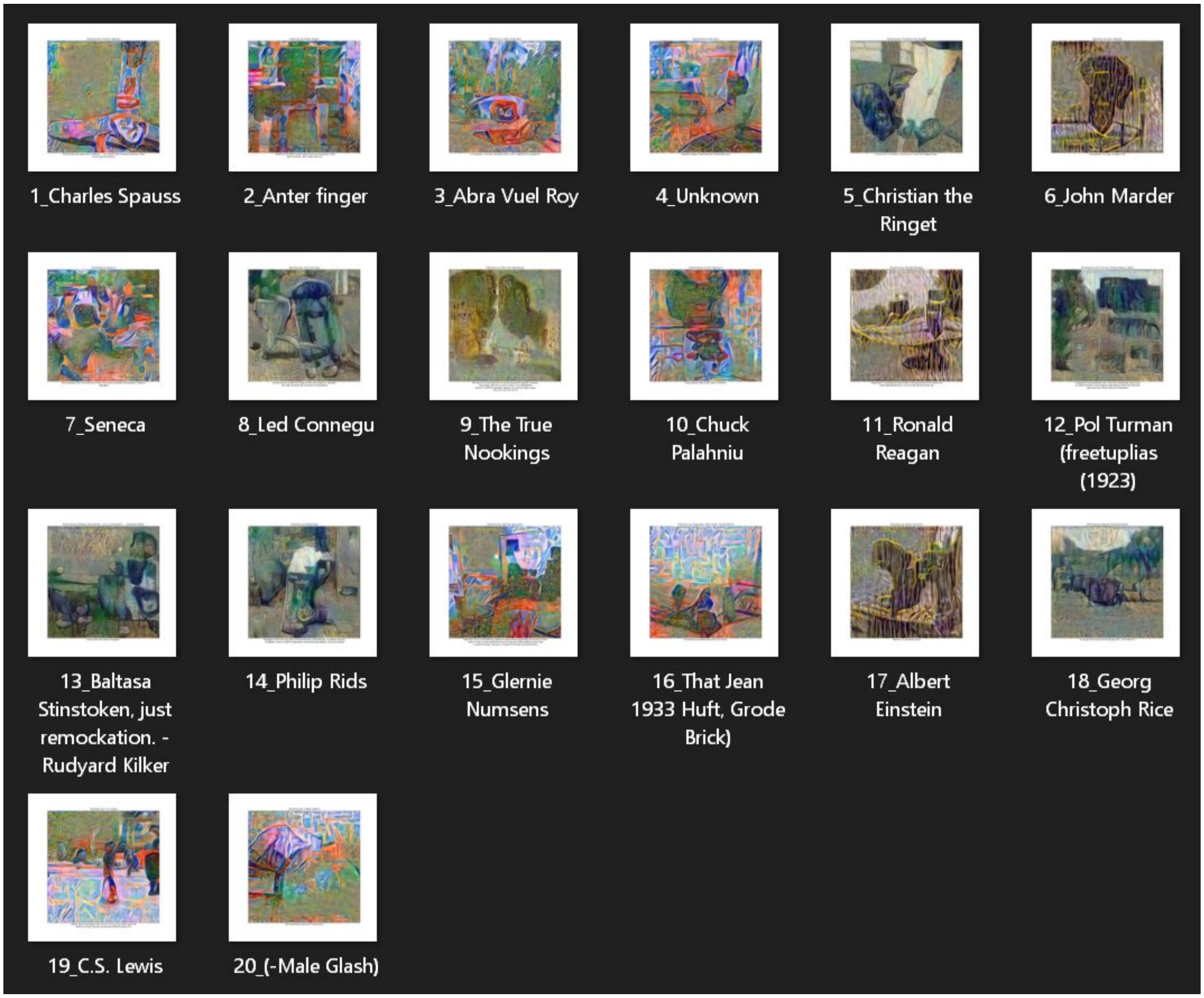
After generating the base image, the sentiment was then used to pick the type of style image that was used for a neural style transfer network. These six sentiments each had their own unique image (where each one appeared to correspond to the sentiment in question). This is where a lot of the interesting part happens as the quote varies wildly enough for different sentiments to show up under different subjects.

PROCESS

At first, I was sure that I wanted to generate something out of text, but none of the networks covered in class dealt with this. So, I decided to look it up online and happened to find AttnGAN, which seemed to be the most popular form of caption to image generation. I thought it would be easy to use given that they had their own evaluation API, but given the not-so-good documentation that was given, it took a much longer time than expected changing the configuration of the evaluation API to fit the pretrained COCO model (that which was provided was only able to be used on the birds dataset). Furthermore, combine this with the fact that the API was web-based (it used Flask to interface with the network), it was a pain dealing with it as it was. So, I decided to scour elsewhere on Github for a more straightforward solution for it; I found it in the form of a non-Flask API after much time.

However, after generating some test images from some quotes I personally made up, I felt that the results (while fairly abstract), still seemed fairly unstylized. That is, it was not reminiscent of much of the artwork I have seen over the years. Convinced that the subject matter was not the problem, I then decided to incorporate the quote's sentiment as a stylistic basis for the image. This part of the project was easy to develop as I already had experience using word-RNN from P1. No hitches were experienced in this part as much of the code had already been attempted throughout the weeks for both of these types of networks.

RESULT



I created quite a few images, but note that a lot of the images ended up being stylistically similar. Part of the reason is that many of the images selected had largely yellow hues associated with them. As a result, it is a little difficult to tell which image corresponds to which input style image. Furthermore, even once you account for this, the images tended to end up with the “joyful” sentiment. This is because quite a few of the quotes that were generated corresponded to entirely padded sentences. These small sentences were then often classified as joyful when put through the word-RNN. That’s clearly a source of bias in the results, but it still generated some results that were satisfying. (This bias should be clear when from about 20 randomly selected quotes almost half were from one sentiment type).

More results can be seen above (note that the text that reports the artist name and quote used cannot be seen in the thumbnail, but they are indeed there in the borders of the image).

REFLECTION

I did not personally select any of the quotes. They were chosen at random so that I could more properly see how the output would generally look like. That being said, I do feel as though the work chosen is fairly interesting. It could be taken much farther than what I currently have by involving a much better text to image generator. Eventually, a dataset that involves even more classes than COCO will be used to generate much more encompassing images. Furthermore, with enough computing power, the type of corpus the sentiment RNN had been trained on can be drastically expanded to something that can perhaps even make a guess as to what type of sentiment an unknown word carries.

Overall, I have learned a lot from this course about many different neural network architectures. While the overview was clearly not too mathematical, it is the experience that is generated from looking into the network and applying it to something that makes it far more useful. Hopefully, this experience will be far more useful in the future when working on ML related tasks.

CODE: https://github.com/ucsd-ml-arts/ml-art-final-ss

[This is accessible through the Github repository]
RESULT: https://github.com/ucsd-ml-arts/ml-art-final-ss/tree/master/Portraits

REFERENCES:
[None were used here, but they can all be seen in the Github repository.]