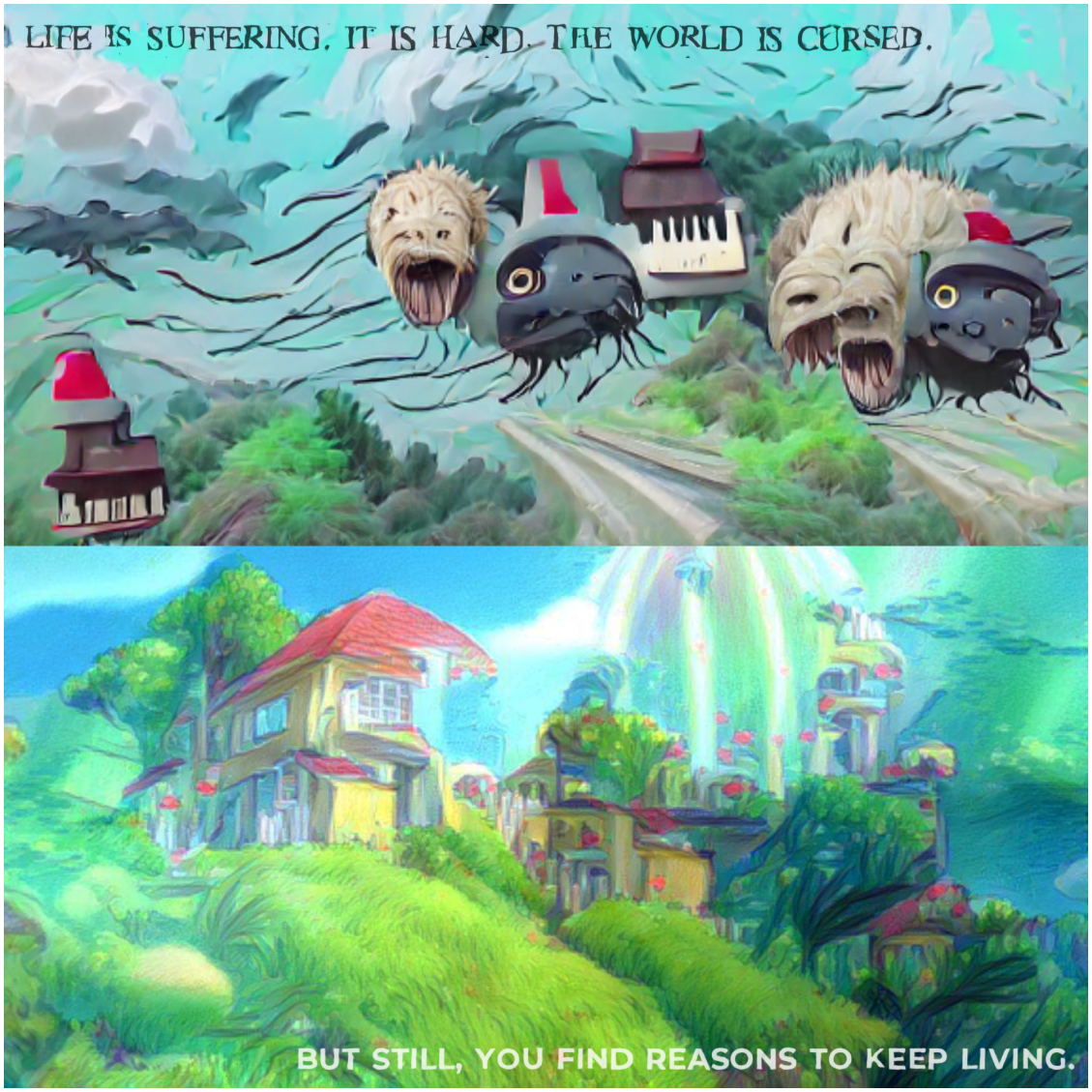
ART AND MACHINE LEARNING
CMU 2022 SPRING
PROJECT 2

The Good, the Bad and the Ghibli



Group 4
Kevin Chen (Master of Computational Data Science, School of Computer Science)
Maya Shen (PhD in Statistics & Data Science, Dietrich College)
Kayo Yin (Master of Language Technologies, Language Technologies Institute)
Kenneth Zheng (Master's in Intelligent Information Systems, Language Technologies Institute)

DESCRIPTION

Inspired by the duality of Studio Ghibli movies, we hope that our work will remind the viewer to consider the duality of the many aspects of our lives and how our perceptions change as we learn and grow. We explored different approaches to generate letters (CycleGAN, Neural style transfer, and a custom feed-forward neural network) and Ghibli-like background images (VQGAN, CLIP, a modified STROTSS, AnimeGAN). In our final result, we juxtapose two generated images and two halves of a quote drawn with our model, with one image creepy and one image pleasant and one half of the quote dark and depressing and the other half uplifting. This artwork will remind each of us that “Life is suffering. It is hard. The world is cursed. But still, you find reasons to keep living.”

Concept

Our work is inspired by the duality of Studio Ghibli movies. Because Ghibli movies are animated and often brightly colored, people often think of them as being cheerful and childish. While some fit that criteria, most of them are actually much deeper and darker than an outsider might perceive.

The two main inspirations for our project were 1. letter/font generation and 2. Studio Ghibli. In the process of group brainstorming, we iterated through many different ideas, but ultimately found the idea of letter/font generation and Studio Ghibli to be the most interesting. We began working on many different methods, some of which were solely Ghibli-style or letter/font generation-related and others which sought to combine the two. Overall, in terms of Ghibli, we were mostly focused on trying to convey the Ghibli style we all loved.

The biggest pivot point for our project was the inadvertent generation of creepy Ghibli images which prompted us to step back a bit. One thing some group members realized was that, while our present views on Ghibli films were largely positive, that was not always the case. In fact, two out of four group members had distinct memories of being terrified by the movie *Spirited Away* as children. We then spent a bit of time thinking about childhood vs adulthood; in particular how, in the process of growing up, it can be easy to become nostalgic and romanticize our childhoods (like we may have done with certain Ghibli movies) and how, as we gain experience, we are able to recognize and understand messages and themes that we may have missed when we were younger (e.g. many of the more serious themes in Ghibli movies).

Given all our discussions about the duality of Ghibli movies and our experiences with them, we decided to move forward with the theme of duality within and related to Ghibli movies. Because we had already generated creepy images and had also independently generated letters that looked creepy given enough added noise, we had the idea to overlay a dark quote using the creepy letters over the creepy Ghibli image to represent the darker themes of Ghibli films and the fear and anxiety they can arouse in

children. Since we wanted to explore the duality of Ghibli, we needed something on the other end of the spectrum, something more peaceful and cheerful to represent our love and appreciation of the films, their artwork and messages, and everything else they have brought to our lives. We chose the following quotation from Princess Mononoke to illustrate the message we often derive from Ghibli films that despite all the hardships and pain we may experience in our lives, each of us has a purpose in this world: “Life is suffering. It is hard. The world is cursed. But still, you find reasons to keep living.” Ultimately, we hope that our piece will remind the viewer to consider the duality of the many aspects of our lives and how our perceptions change as we learn and grow.

Technique

For the text, we used a custom neural network trained to draw letters. We were inspired by this neural image painting demo [7], which showed that an image can be thought of as a function from (x,y) position to (r,g,b) pixel values. This means that a simple feed-forward neural network can be trained to model this function as a regression problem and “paint” the training image. The resulting network can then be used to generate images of arbitrary size, since the (x,y) position space is continuous. The results also happen to be quite aesthetically pleasing with a painterly style. We also took inspiration from this blog post [8], which extended this idea to include an additional latent vector as a conditional input to the network to generate abstract images. This type of neural network is sometimes referred to as a Compositional Pattern Producing Network (CPPN) following [11].

To apply this model to generate text, our neural network takes two inputs representing the (x,y) position in the image, and an additional vector which represents a one-hot encoding of the letter to draw (so for all 26 uppercase letters, this vector would be length 26, with a 1 in the first position indicating ‘A’). The network architecture consists of 8 hidden layers of size 20 using tanh activations, and a final output layer which maps to 3 (r,g,b) outputs and clamps them to a range of [0, 1] using a sigmoid function.

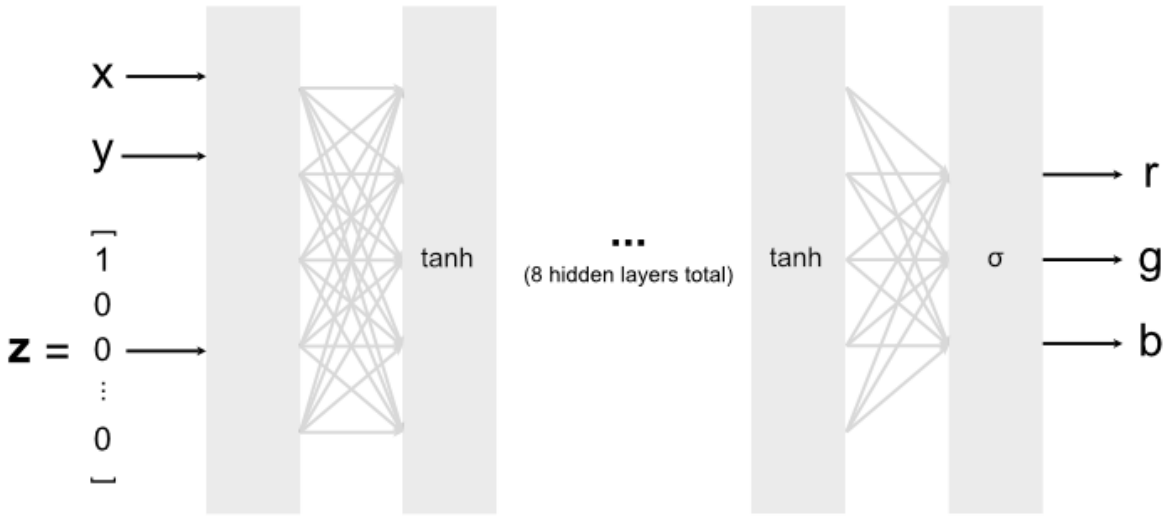


Figure 1: Model architecture diagram

To train this network, we extract images of characters from a font, resize them to a height of 200 pixels, and convert each pixel into a training example with (x,y) position (rescaled to [0, 1]) as the input and (r,g,b) values (also scaled to [0, 1]) as the label. The model is trained to minimize the mean-squared error loss using the Adam optimizer. Once trained, to generate each text character, we feed in each pixel position and the one-hot vector representing the character to the model. To add variation to each character, we add a small amount of Gaussian noise to the one-hot latent vector which creates some interesting results. Figure 2 shows the effect of different noise amounts when generating text. Coincidentally, adding more noise seems to have the effect of making the characters look more creepy, which was the aesthetic we were going for in our final piece.

Figure 2: Effect of different noise levels on generated text

For our final image, we used two models to generate the two quotes, one trained on the font Times with more noise for the creepy text, and one trained on the font Montserrat with less noise for the peaceful text. We also explored some unrelated tasks for our model such as generating new characters that are interpolations between letters, which we describe later in the Process section of this report.

For the background of the creepy image, we used VQGAN+CLIP to generate images based on text prompts. We will go into more detail on the text prompts explored in the following section. VQGAN+CLIP is a combination of two neural network architectures: Vector Quantized Generative Adversarial Network (VQGAN) [1] and Contrastive Language-Image Pretraining (CLIP) [2]. VQGAN is a GAN that combines convolutional neural networks and Transformers, and it is good at generating images that look similar to each other. On the other hand, CLIP is a model trained to determine which image caption from a set of captions is best suited for a given image. The two work together in VQGAN+CLIP with CLIP guiding VQGAN towards an image that best matches the given text prompt.

For the background of the relaxing image, we used a modified version of Style Transfer by Relaxed Optimal Transport and Self-Similarity or STROTSS, an optimization-based style-transfer algorithm [6]. By modifying the STROTSS Colab notebook by Peter Schaldenbrand and the code from David Futschik's github (which was also the code used in the aforementioned Colab notebook), we were able to create an algorithm that uses two style images instead of one; let's call this method Two-STROTSS. More specifically, we overwrote the `calculate_loss`, `optimization`, and `strotss` functions with our modified versions which take in two style images and have differing image initializations. Originally, we had created this method to generate stylized letters (see Process section for more information on these explorations), but were able to successfully use this method to generate pleasant Ghibli-style backgrounds.

After selecting our creepy and pleasant images, we spent some time looking for quotes that would fit our project and the two images. We narrowed it down to wanting to use quotes from Studio Ghibli movies or Hayao Miyazaki. Ultimately, we chose the quote "Life is suffering. It is hard. The world is cursed. But still, you find reasons to keep living." from *Princess Mononoke*. To put everything together, we manually combined our two background images and generated quote images in an image editing program.

We utilized a great number of other tools throughout this project which were not used directly in the final product itself. However, most of them are described and referenced in the following Process section.

Process

Letter Generation

To generate Studio Ghibli-inspired letters, we first tried training CycleGAN [3]. CycleGAN is a GAN that performs image-to-image translation, and its key advantage is that the architecture does not require paired data during training. For the source data, we used stills from Studio Ghibli (https://www.ghibli.jp/info/013344) that has been masked out to form capital letters in Times New Roman (Figure 3; left) and for the target data, we used the 26 illustrations by illustrator Ekisha Narain (https://dribbble.com/ekisha/shots) of Ghibli-inspired letters (Figure 3; right).

Figure 3: Example of source and target images used to train CycleGAN

In the following image, we see examples of an input to CycleGAN, the output in the other domain, and the reconstruction in the original domain:

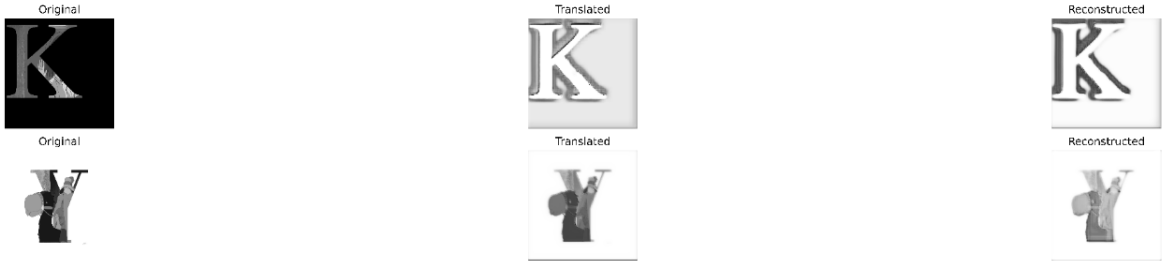


Figure 4: CycleGAN's poor results trying to translate images into the other domain.

As we can see above, CycleGAN does a poor job of translating images into the other domain. We abandoned this direction as other directions gave more promise.

Next, we tried using neural transfer [4] to generate the letters filled with style-image as background. This techniques don't have to save the model but it takes around 20 mins to generate a style-transfer letter. We tried a couple different methods. One such was exploring different combinations of layers and weight to generate the letter images. However, we couldn't get a good result at first. We then tried out different layers to design our loss function and eventually created some cute letters. In order to generate

the letter with a transparent playground, we use original letter images to mask out the black region to make it transparent. Thus, our final Ghibli-style letters can be easily put onto other images. We eventually didn't use this font since we want to generate the creepy-style image as our final result instead.

Figure 5: Neural transfer to generate stylized letters.

Another method we tried for generating stylized letters was style transfer: STROTSS and our modified version of STROTSS, Two-STROTSS. We started out initially with STROTSS; more specifically, we were trying to transfer the style from a single Ghibli still to a letter. We decided on Times New Roman as our font of choice for this exploration method. Initially, we ran STROTSS on letters with black font and white background. However, we realized that, while the resulting images were nice, it did seem that the area within the resulting letter was dark while the edge around the letter was light. We thought that the inverse of that (i.e. having the area within the letter light and the edge around the edge dark) could be a more striking look. Thus, we tried the inverse of the original letter: white font and blac background and achieved the look we were aiming for. We ran STROTSS in this manner on many Ghibli stills; two, both from *Princess Mononoke* are shown here in Figures 6 & 7.

Figure 6: STROTSS style-transfer with style from *Princess Mononoke* to the letter A with white font and black background and black font and white background with content_weight = 0.3.

Ultimately, we decided that we preferred the look of the result of STROTSS on letters with white font and black background over the inverse. Next, we investigated the effects of the parameter content_weight for STROTSS. We tried out different content_weight values for multiple Ghibli stills and found that we liked the results from content_weight values around 0.3. We show one set of results in Figure 8; note that the results for content_weight values 0.1 - 0.5 look approximately the same, the image from content_weight = 0.8 looks the most different and blurred.

Around this time, we had the idea to utilize multiple style images instead of just one. Initially, the idea was inspired by a group member's curiosity about the possibility of learning an artist's overall style and transferring that as opposed to transferring the style of a single piece. Naturally, the first step towards that is two simply try to transfer two styles at once. We tried many different combinations of Ghibli stills while varying the content_weight parameter slightly. Figure 9 shows some of such explorations.

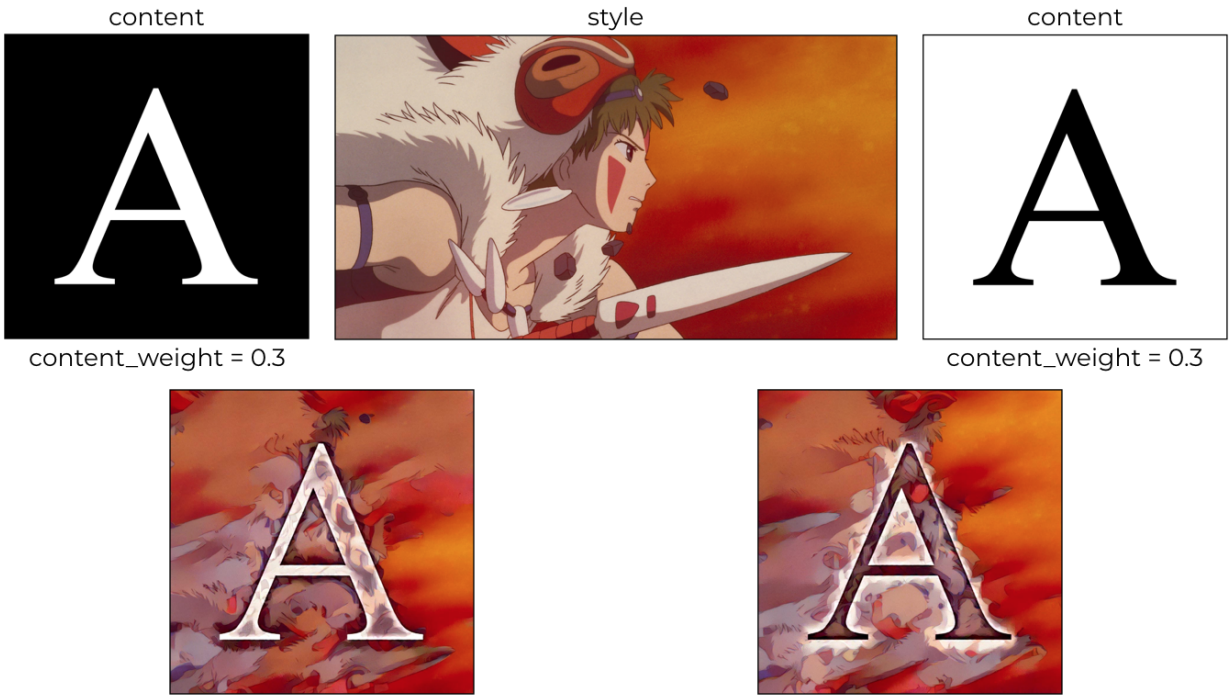


Figure 7: STROTSS style-transfer with style from *Princess Mononoke* to the letter A with white font and black background and black font and white background with content_weight = 0.3.

Some other avenues in this area we explored were changing the letter's (content image) font and background colors, different image initializations, and (accidentally) using huge values for the content_weight parameter. More specifically:

- We were interested in using the most "common colors" within the style images as the font and background color of the content image (letter). We did this using k-means to cluster the colors within each style image into 10 clusters, selecting the largest cluster, and then taking the average color of that cluster.
- The original image initialization within the STROTSS Colab notebook takes the sum of the Laplacian of the content image and some averaged values from the style image. To convert STROTSS to Two-STROTSS, we simply added the averaged values from the second style image. Another initialization method we tried was to begin with random noise.
- We accidentally inputted very large values (e.g. 10, 50, 80) as content_weight parameters which resulted in brightly colored noise images with white letters.

Ultimately, we did not find anything concrete from these three avenues during our exploration but we think the first two would be interesting to work on in more detail.

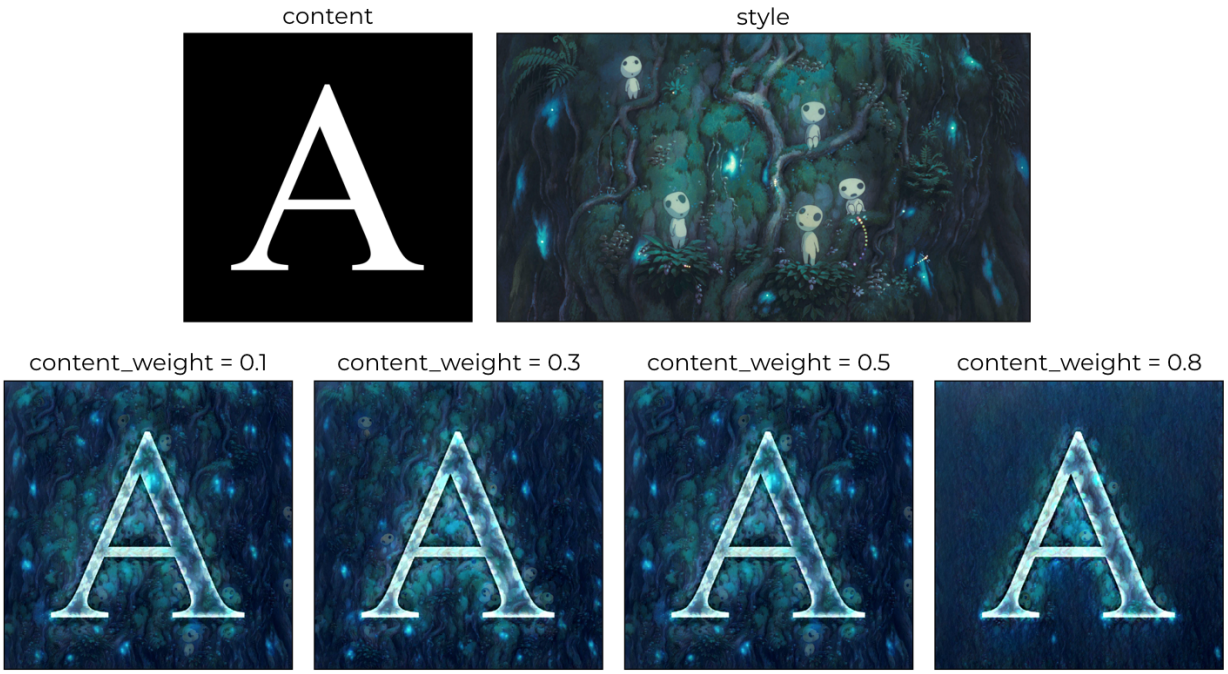


Figure 8: STROTSS style-transfer with style from *Princess Mononoke* to the letter A with white font and black background for content_weight values = 0.1, 0.3, 0.5, 0.8.

Figure 9: Two-STROTSS style-transfer using two Ghibli stills as style images and the capital letter A in white font with a black background as the content image along with varying content_weight parameters.

Finally, we also tried using a backpropagation method inspired by DeepDream [9] to get a model to hallucinate letters in an image. We trained a simple convolutional network on EMNIST [10], a dataset of handwritten characters, and then backpropagated from the output where a single letter label was high, using this gradient to modify the input image. While it did kind of work (you can see the letters if you look for them), it was overall too noisy and low-resolution to pursue further.

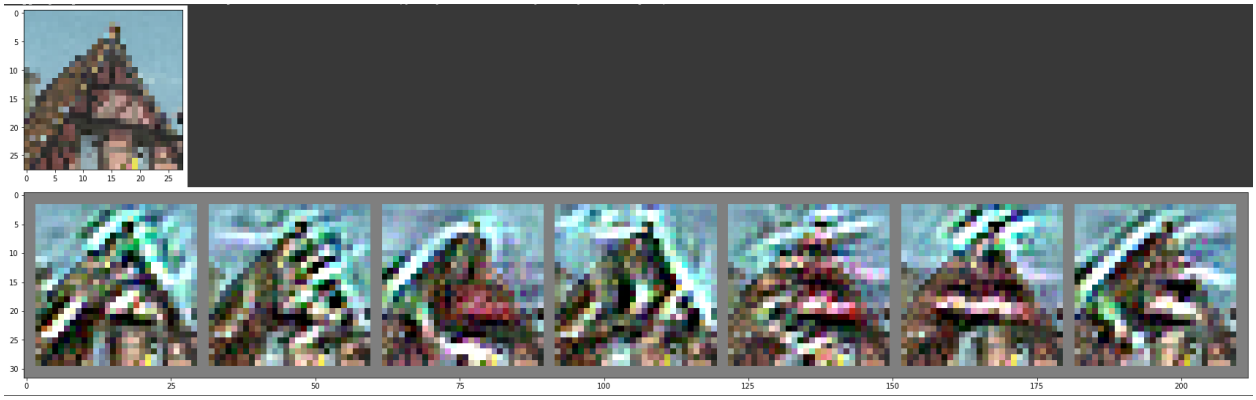


Figure 10: Backpropagation using an EMNIST classifier, source image above, hallucinated letters A-G below

Overall, while many of these text generation explorations produced interesting results, we eventually settled for using the feed-forward network approach, described in the Technique section, for our final composition.

Background Image Generation

For the background image, we first experimented with AnimeGAN:

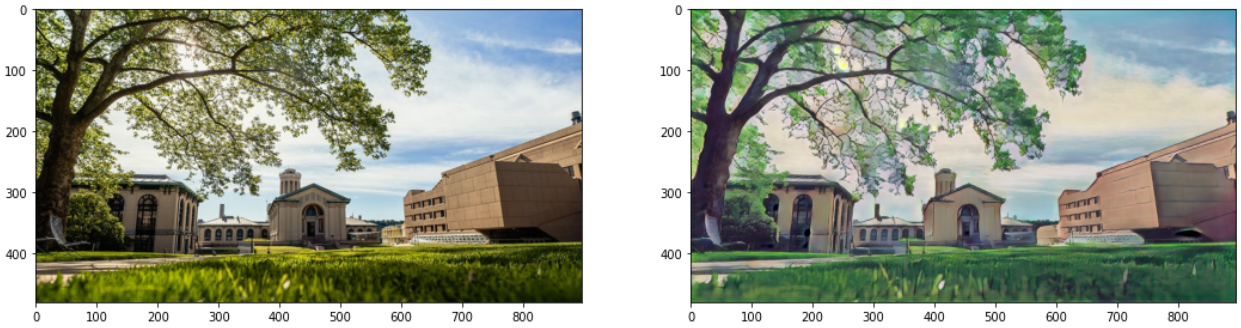


Figure 11a: AnimeGAN results using a picture of CMU

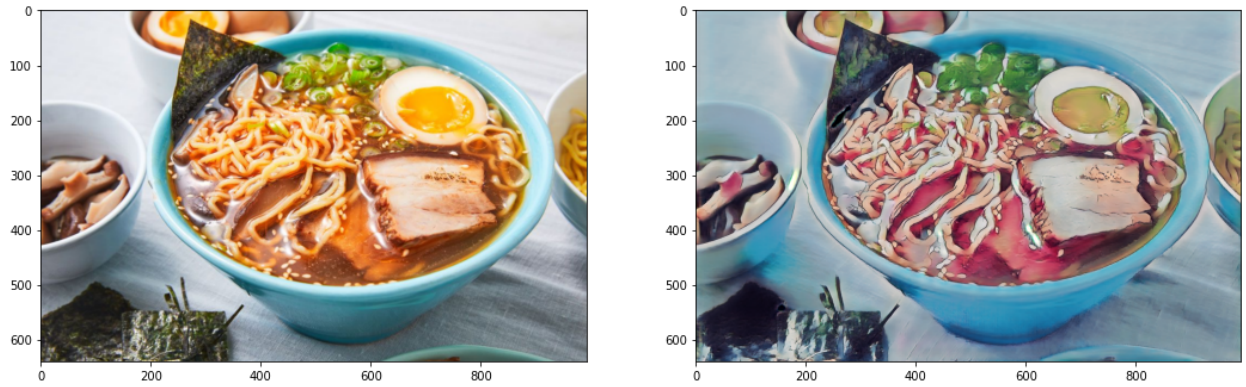


Figure 11b: AnimeGAN results using a picture of ramen

Next, we tried different text prompts to VQGAN+CLIP. We let the model run for 400 iterations, which takes about 40min for each image. We detail the different prompts we explored as well as the resulting output below.

Prompt: a castle floating in the sky in the style of studio ghibli
Image:

Prompt: a green forest with small flowers that looks like the face of a beautiful woman in the style of ghibli
Image:

Prompt: several fantastic creatures in the style of ghibli melting together
Image:

Prompt: several creepy screaming creatures in the style of studio ghibli
Image:

Prompt: scary creatures flying on a landscape in the style of studio ghibli
Image:

Figure 12: Creepy images generated using VQGAN+CLIP.

After inadvertently generating the creepy Ghibli images and deciding to pivot to the duality of Ghibli, we still needed to generate peaceful and pleasant images. We wanted to try to use Two-STROTSS to generate such an image. However, we were unsure what the content image should be; Figure 13 shows two of our early attempts which were not very successful. One content image was a photograph of mountains and the other was a drawn landscape of mountains. Both results were interesting but not especially pleasant.

Figure 13: Initial Two-STROTSS style-transfer attempts using a photograph and a hand-drawn drawing of mountains as content images and Ghibli stills as style images.

Figure 14: Two-STROTSS style-transfer results using solid white image as content and Ghibli stills as style images, two runs for each set of style/content images.

Next, we had the idea to simply have the content image be solid white and the content_weight be 0.0. This actually returned rather pleasing results. Two of the nicer sets are displayed in Figure 14. Both were run twice to return the two similar but different results in the bottom right of each set. Ultimately, the top result from the right set was chosen to be our pleasant image.

Other Explorations

Before using our custom neural network to generate letters, we first tried using it to generate images like in the original demo. One interesting discovery is that the choice of activation function changed the overall aesthetic of the generated image in a predictable way. ReLU activations create sharp edges and angles, while tanh activations create soft edges and rounded shapes.

Figure 15: Effect of activation function choice on neural “painting”, tested on a frame from *Princess Mononoke*

Another interesting thing we explored was what would happen when we tried to interpolate between letters in the latent space, by setting both the indices of both letters as 1 in the input. Below is a complete grid of interpolations for each letter combination.

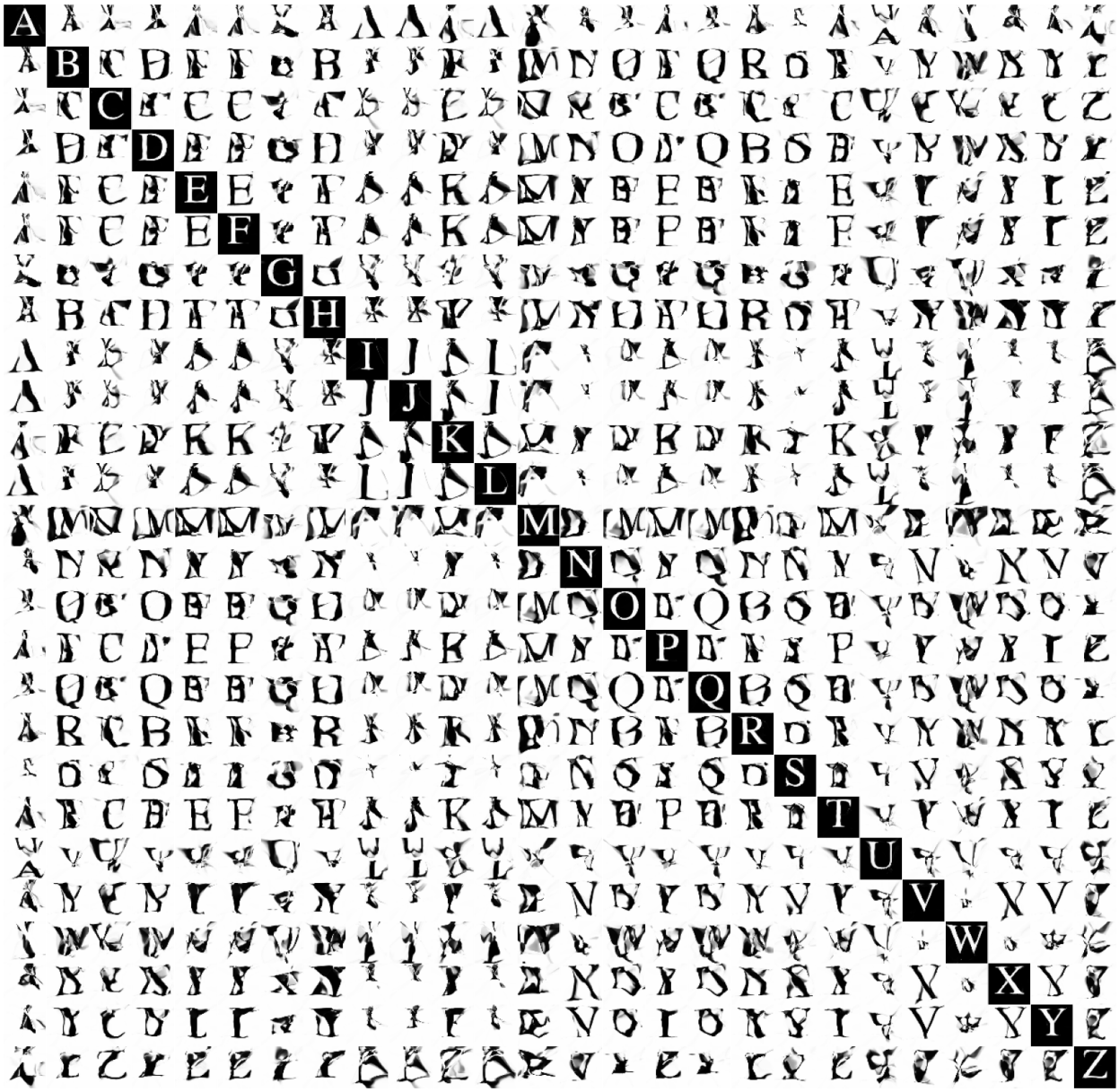


Figure 16: Interpolations between letters that our model generates (regular letters inverted for visual clarity)

Finally, here are some completely random samples from the latent space, which are also very interesting and surprisingly artistic.

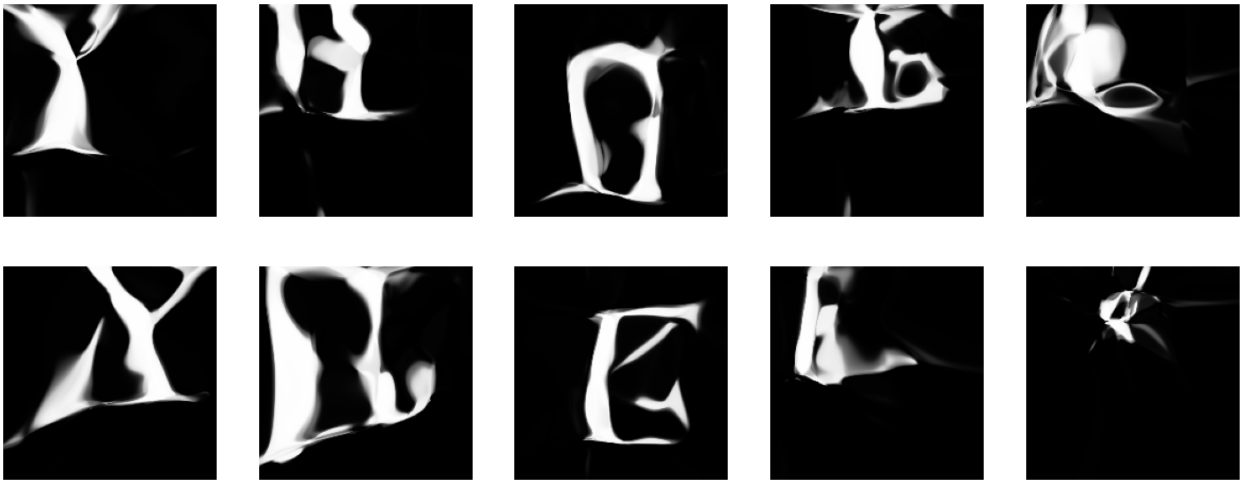


Figure 17: Some randomly sampled model outputs (inverted to show detail)

Reflection

For text generation, we initially thought that the CycleGAN architecture would be promising to generate pretty Ghibli-style letters, due to the power of the model and its ability to learn from unpaired data. However, we found out that the results were very weak and did not provide what we expected. We learned that this is due to too much mismatch between the two domains, which leads to poor reconstruction loss and inhibits the ability of the model to learn the target domain. Although such architectures work well for, say, making a horse look like a zebra, it is not suitable for more complex tasks where there are more features the model has to learn.

The method of neural style transfer applied to letters was more promising, and generated some interesting stylized letters. However, as we wanted to display many letters in the form of a quote in our final composition, these methods were ultimately too time-consuming to run and didn't lend themselves well to a cohesive final result.

Our final text generation model was the simple feed-forward neural network, trained to draw each letter based on a pixel position and a conditional latent vector. Although this model is fairly simple, the addition of the latent space enabled the model to have lots of expressive power, and the continuous nature of the position inputs means that the model can generate images of essentially infinite dimensions with a lot of detail. For our final result, we chose to use the latent space to add noisy artifacts to our text, making each generated character unique in a way that would only be possible with such a machine learning method. Additionally, we further explored the latent space, seeing what the model thought "interpolations" between different letters would look like, as well as simply randomly sampling from the latent space which created cool abstract patterns. This type of network definitely has

many other artistic applications (e.g. modifying network structure, activation functions, and/or training image sets), which we are interested in exploring further in the future.

For generating Studio Ghibli-style backgrounds, we thought that AnimeGAN would be one of the most viable options, as the network has been trained specifically to imitate the style of Hayao Miyazaki. Although AnimeGAN indeed does a good job at recreating anime scenes from photos that are faithful to the style, the outputs were still unsatisfactory for our project. We wanted the contents of our image to be more creative, whereas AnimeGAN can only output the same content as the input photography. We therefore opted for image generation algorithms that can output more creative images.

We started using VQGAN+CLIP to generate beautiful images in the style of Studio Ghibli. While the model seems to have learned some world knowledge and is able to reproduce Ghibli-style images, when given the second prompt, it generated an image that was unintentionally creepy. This process has led us to remember that Studio Ghibli films are not as feel-good and wholesome as they seem at first glance: they often deal with darker and more serious issues such as war, environmentalism, and identity. And yet, this complexity gives viewers a lot of satisfaction and “feel-good”, precisely because it reminds us that despite all the suffering that life brings us, life is still beautiful. This was the turning point of our project where we decided to explore the theme of duality in Ghibli, and we chose to create both images with a lighter and a darker theme. By using this model and tuning the text prompts, we were able to generate a creative image that reflects the mood that we wanted to convey (creepy, anxiety, strange) while remaining in a style reminiscent of Studio Ghibli animations. We used the last image generated for the creepy background, because of the bright colors that are faithful to Studio Ghibli, the strong dark lines that give a dark edge to the colorful image, and the strange creatures with gaping mouths and big round eyes floating around that induce anxiety in viewers.

The other method we explored was attempting style transfer using multiple images. We were originally interested in style transfer to try to generate Ghibli stylized letters. One of our ideas was to extract or learn the style from a set of images and apply that to the letter images but ended up sticking to using two images with a simple modification to the STROTSS loss function to obtain Two-STROTSS. Still, the combination of two images was enough to create some artistic results, one of which we used in our final piece. While we ended up moving in a different direction, seeing what parameters and style images generated interesting stylized letters and combined images sparked interesting ideas on ways to explore that path further. We covered some briefly in the Process section but they are as follow: 1) how the letter image’s font and background colors or the initial blank content image’s color could affect and potentially enhance the final result, 2) other image initialization methods both random and non-random, and 3) a Multi-STROTSS algorithm which implements the original idea of learning a style from a set of images.

Additionally, generating stylized letters which captured some of the beauty and magic of Studio Ghibli artwork pushed us to think a bit about how one would/should display such results. Many of our ideas revolved around childhood and learning the alphabet and how to read, so things such as a poster or book with the alphabet made up of stylized Ghibli letters where each letter stood for a Ghibli subject and was generated using images of it, e.g. P is for Ponyo with the P generated using stills of Ponyo. While

considering the topic of childhood, we also thought a bit about how prevalent AI/ML is in our everyday lives and how children nowadays are growing up with AI/ML as part of their lives and growing process. In fact, a children's alphabet book with stylized letters generated using ML techniques may be a good metaphor for that.

We are proud of both our final composition and all of the other experiments we performed. Although most of our work didn't make it into the final result, we learned a lot from the process and gained insight into many different forms of machine learning for image generation and visual art.

RESULT

The full-resolution file can be found here: [Group4_TheGoodTheBadAndTheGhibli.png]

Our final result is meant to be a representation of the duality in Ghibli which can often be found in life as well. In our final piece, we juxtapose two images and two halves of a quote, one image creepy and one image pleasant and one half of the quote dark and depressing and the other half uplifting. The top half is

meant to unsettle the viewer, both in the creepy image and the depressing quote. The creepy image contains strange floating creatures (many would call them monsters) and unsettling dark lines that may make viewers feel anxiety and claustrophobia, while the quote speaks to the hardships of life and the darkness of the world as a whole. But when your eyes then shift to the bottom half, we see a bright and cheerful scene, like something out of a fairytale: A yellow house on a grassy rolling hill with beams of light shining down and perhaps a bit of an ethereal underwater feel.

The quote itself is from the Ghibli movie *Princess Mononoke*, drawn with two of our custom models for text generation. While the two images were generated separately and using different methods, they were both generated using some relation to Studio Ghibli – the top creepy one with text prompts that included the phrase "in the style of studio ghibli" or similar and the bottom pleasant one with stills from Ghibli movies. Thus, there are actually some similarities (such as in the significant presence of blues and greens) between the two images despite how different of feelings they invoke in us. Additionally, those who are familiar with Ghibli will likely see aspects in the creepy image that they can connect to creatures and scenes from the movies. Perhaps the gray floating beetles have some relation to the Ohmu in *Nausicaä and the Valley of the Wind* or the buildings are taken from the spirit village in *Spirited Away*? The bottom image is much easier to place as it was created using two beautiful stills from *Ponyo*: one of Sosuke's house on the hill and the other of Ponyo's underwater home where her sisters still live. Despite this image being a result of merging two wildly different worlds, the result does not appear out of place or strange. In fact it seems to make a lot of sense…

CODE

https://github.com/kayoyin/ghibli-ml-art

REFERENCES

[1] Esser, P., Rombach, R., & Ommer, B. (2021). Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12873-12883).

[2] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning* (pp. 8748-8763). PMLR.

[3] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223-2232).

[4] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge. ex "A Neural Algorithm of Artistic Style".

[5] Chen, J., Liu, G., & Chen, X. (2019, November). AnimeGAN: A novel lightweight gan for photo animation. In *International Symposium on Intelligence Computation and Applications* (pp. 242-256). Springer, Singapore.

[6] Nicholas Kolkin, Jason Salavon, Greg Shakhnarovich (2019). Style Transfer by Relaxed Optimal Transport and Self-Similarity. CVPR.
[7] Kaparthy, Andrej. ConvNetJS demo: Image Painting.
https://cs.stanford.edu/people/karpathy/convnetjs/demo/image_regression.html
[8] Generating Abstract Patterns with Tensorflow (blog post).
https://blog.otoro.net/2016/03/25/generating-abstract-patterns-with-tensorflow/
[9] Mordvintsev, A. & Tyka, M. (2015). Inceptionism: Going Deeper into Neural Networks. Google AI Blog.
[10] Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters.
[11] Stanley, Kenneth O. (2007). Compositional Pattern Producing Networks: A Novel Abstraction of Development. In *Genetic Programming and Evolvable Machines, Special Issue on Developmental Systems.* Springer.