Bloom: Blurring the Lines between Human, Plant, and Machine

*Eesha Shetty *, Lia Coleman*, Michelle Zhang*, Yvonne Fang**

1. Description

This project aims to create a system that combines sensory elements with image generation using StyleGAN to study the concept of **latent space interpolation**. The end goal is to generate a GAN image that seamlessly interpolates based on both electric signals from the plant and sensory input obtained from interacting with plants.

2. Concept

The project will involve collecting sensory information associated with different plants, training a StyleGAN model on this dataset, and exploring the latent space to manipulate the latent vectors to generate a wide range of plant images

This is an interactive art project that explores co-creation between humans, plants, and AI through touch-based input. By using sensors placed on a plant's leaves and stems, participants can collaborate with a living organism to create a real-time generative output of StyleGAN generated visuals of a plant's leaves, which are influenced by both the plant's biological signals and the participant's touch. Inspired by the art project presented in class where Hubert Duprat collaborated with caddisfly larvae to make gold-leaf cocoons [1], this project blurs boundaries between human, plant, and machine to create an interactive performance.

Another precedent we found was Intelligence all the way down [10] where the artist explored biological processes as a non-human form of intelligence, which is similar to signals we get from plants' internal biological processes. Instead of trying to make sense of the signals we get from the plant, we allow it to control the latent space interpolation of StyleGan3 fine-tuned on our leaves dataset, giving the plant agency in collaborating with the model, and even humans, to create art.

3. Technique

3.1. Fine Tuning StyleGANV3

We finetuned StyleGAN3 using a plant leaf dataset from the instagram @alongletter [2], which consisted of 514 photos of leaves taken from the same fallen tree [Figure 1].

We started with the FFHQ model and trained it on Google Colab. We picked a few model checkpoints that we liked the output images from– we especially liked a few that had some unexpected but beautiful artifacts from the data augmentation process during training– one which appeared smoky, and another which appeared neon. [Figure 2]

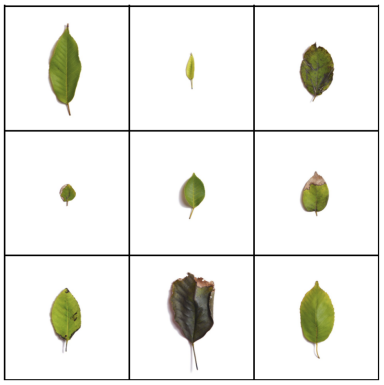

Figure 1: *Leaves Dataset*

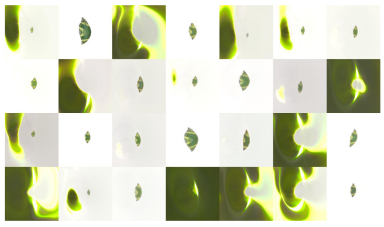

Figure 2: *Generations by our fine tuned model*

3.2. Latent Space Interpolation with StyleGANV3

With StyleGAN, we can generate images from a latent code. These latent codes can be generated randomly given a seed. Once we generate two latent codes, we can generate a seamless combination of both by exploring the concept of linear interpolation. The linear interpolation formula is the simplest method used to estimate the value of a function between any two known points. We use this in the context of latent codes, to interpolate between two and generate our own latent code which still fits the generative distribution.

By setting a value alpha, we can decide how much of either seed we want the final image to look like.

- **If alpha = 0**, the generated latent code would be exactly as the second latent code provided

*Equal Contribution

All authors are Masters students at Carnegie Mellon University, this report is a part of our final project for our Art and Machine Learning Class.

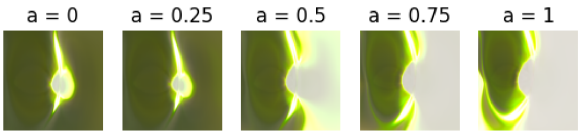

Figure 3: *Generated Image at Different Alphas*

- **If alpha = 1**, the generated latent code would be exactly as the first latent code
- **If alpha = 0.5**, the generated latent code would be an image exactly between the two latent codes

We followed a blogpost on how to use StyleGAN's generative functions to generate latent codes and then fit the linear interpolation formula. The biggest challenge with the code in the tutorial was that it was based on StyleGAN V2, its code base is written in Tensorflow 1 and Colab no longer has support for this package. So, we decided to migrate to StyleGAN V3, which is written in PyTorch so it was much more stable and recent, and used the ideas from the blogpost [12] to implement all the functions accordingly.

The end result of the project was a set of functions that can

- Generate images given a latent code
- Generate images given a seed (the seed would randomly generate a latent code)
- Generate an interpolated image between two latent codes given alpha (this would create an interpolated latent code and then use function (1) to create the interpolated image).

3.3. Getting Signals from Plants

To get the signals from the plant, we use an Arduino implementation of Disney Research Lab's technology Touché [3]. The Arduino implementation was largely adapted from this repository [4], the hardware is adapted from this instructable [5]. Touché performs a frequency sweep to collect data on changes of capacitance, which changes as we touch the object being sensed in different places. This frequency sweep looks like the graph in Figure 4 - the line changes when you touch the object being sensed in different places.

We create an alpha value from this line graph by taking the difference of two time steps (subtracting two arrays to create a difference array). We then take the mean of this difference array, to create the final alpha value to send into the StyleGAN. Another method we tested was just taking the y value of the peak in the above graph and then mapping the range of that value to the range of 0 to 1, and then sending that as our alpha. We found that this latter method worked better in the end, but many other methods could also work.

A sample setup of the hardware can be seen in Figure 5. The communication between Arduino and model goes through Processing for graph visualization (also because there is a checksum implemented between the Arduino serial and Processing serial to make sure the data is not corrupted). From Processing we do the difference array calculation, and send the final alpha value to the python program where we're running inference through UDP sockets. This then controls the latent space interpolation.

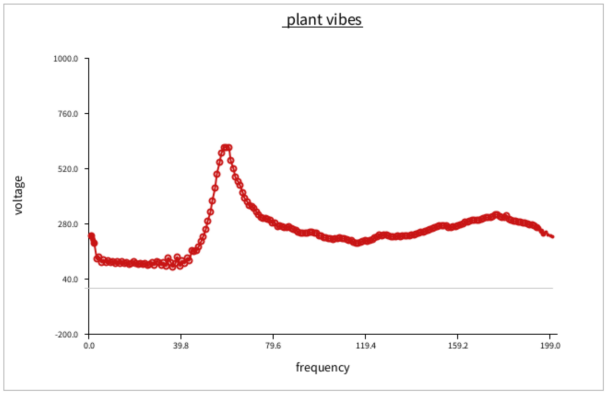

Figure 4: *Frequency Sweep*

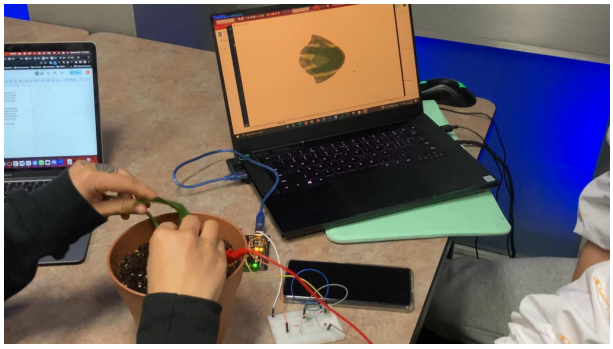

Figure 5: *Sample Hardware Setup*

4. Process

In the brainstorming phase, we knew we wanted to make something interactive and also realtime, since we could build off of work from the previous project where we had realtime generations from a model in TouchDesigner. Conceptually, we also started off with ideas related to haptic/touch input and working with something biological, like a plant, to create some AI-generated output (either music and/or visuals). We were inspired by the recent discovery that plants make popping noises when they're distressed [6], to try to listen into the secret signals of a plant and translate it into some human-perceptible output.

Initially, we experimented with Pix2Pix following this tutorial[8] since the method is to generate real time visuals from a visual representation of the soundwave, which is similar to the plant signal data that we will get. We trained the Pix2Pix model on some nature landscape and the soundscape associated with it, and got some interesting results (below). However, we did not proceed with Pix2Pix because we wanted to explore some other options that can more directly translate sound/music to visuals.

We considered using Riffusion [7] to generate music, but then changed after realizing that diffusion model inference would be too slow for us to run real-time. So we switched to using GANs for real-time inference, and decided to use the input signal from the plant to interpolate between two seeds of a GAN's visual output.

To finetune StyleGAN3, we used a plant leaf dataset from the instagram @alongletter [2], which consisted of 514 photos of leaves taken from the same fallen tree.

We finetuned StyleGAN3 on this dataset, and found some interesting visual effects that popped up during training which appeared between normal-looking leaf outputs. This may be due to data augmentations that happen during the StyleGAN3 training process. However, we were pleased with these results, which were more expressive and painterly.

We ran into some difficulties when trying to adapt our existing TouchDesigner setup to StyleGAN. The original setup in TouchDesigner was for a pix2pix model and used Spout to send image data to pix2pix, and Spout again to receive the pix2pix model's output. However, we had float data as an input to the model, and so Spout– as a tool to pass visual data between applications– would be unable to receive this.

The first approach we tried was taking the incoming float data from an ELEGOO UNO 3 communicating over UDP as input to Touchdesigner. We had to figure out how to call StyleGAN's interpolation function from TouchDesigner, which required finding out how to run a python script from TouchDesigner and resolving conda environment path errors, and import errors that we could not resolve. We also tried to use OSC instead of Spout, for the method that float data was sent from TouchDesigner to the model.

Eventually, we decided to bypass TouchDesigner in the initial step, and instead pass the UDP input directly to the python script to run the StyleGAN image interpolation.

However, we did manage to send the generated image and animation loop to Touchdesigner via Spout by downgrading to Python 3.7.6 since Spout only supports up to Python 3.7. [9] This affords us with the possibility of further manipulating the generation outcome with data such as the plant signal data, or any kind of audio signals.

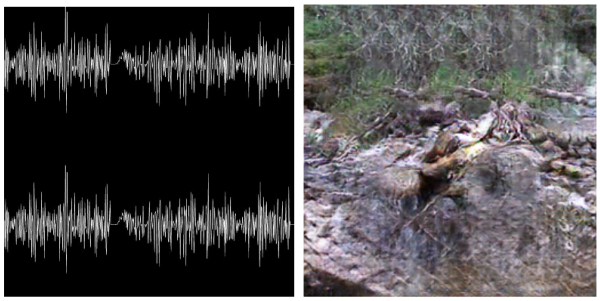

Figure 6: *Pix2Pix Data*

5. Reflection

We found it trickier than expected to wrangle dependencies in TouchDesigner. In the end, we would love to use TouchDesigner so it can all be integrated within it so we can also do post processing on the output if we wanted to.

Also we hope to increase the refresh frame rate of the generations, to make the frame generation smoother. This may be due to the UDP input, as well as the GPU generation power. We plan to swap out the UDP connection that may be slowing

Figure 7: *Training Process*

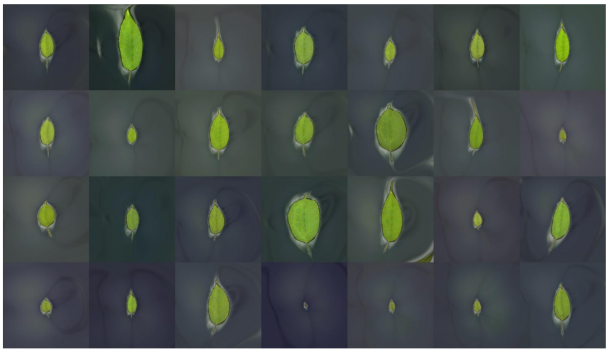

Figure 8: *Training Process*

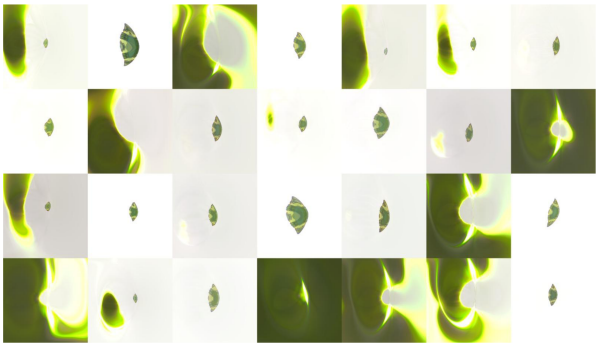

Figure 9: *Training Process*

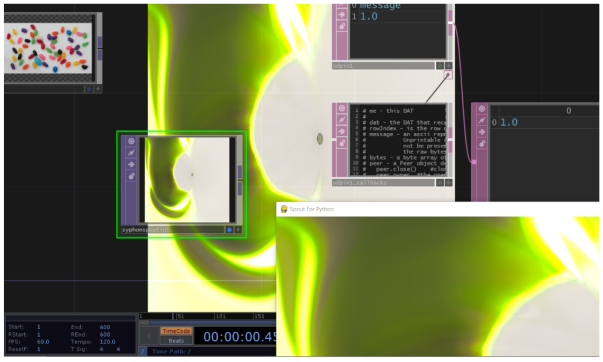

Figure 10: *TouchDesigner Setup*

our frame rate down due to packet loss, to something like OSC or WebSockets.

Additionally, our current project setup interpolates between two chosen seeds produced by the GAN - in future iterations, it would be interesting to explore a more "infinite" search space, maybe by also changing the seeds using the plant data. Another way to make this more varied would be also to use the plant data to drive the interpolation in a different, more expressive way, since the frequency sweeps provide many arrays of values to work with, and it's theoretically able to capture many nuanced gestures (at least according to the research paper).

Regarding the fine-tuned Stylegan3 model, we would like to include more training dataset if possible and if resources allowed. Some motifs we were thinking about include vines, or human hands. We were also thinking about projecting the generation output on a human face or body.

In general, we all really enjoyed working with an organic component, and we're all interested in the sort of collaboration between something organic and something digital. This is definitely a theme that we'd explore further.

Final video demo can be viewed here - https://youtu.be/i5B3xEC-n-k.

6. Code

Github repository:
https://github.com/y1vonnef/stylegan3_artml_final

Latent interpolation:
https://colab.research.google.com/drive/1y_5WAc99sETDZxxmsbCMB31pYsCodHwx?usp=sharing

Leaves dataset:
https://drive.google.com/file/d/182cx2cJir1eAAEkQYAxmC0z3BaVuBfU0/view?usp=drivesdk

For fine tuning StyleGAN3:
https://colab.research.google.com/drive/1LccY-dAAA8JjNeqr40zRLqBEl0wVm6U5?usp=sharing

7. References

1. https://www.thisiscolossal.com/2014/07/hubert-duprat-caddisflies/
2. https://www.instagram.com/alongletter/
3. https://la.disneyresearch.com/wp-content/uploads/touchechi20121.pdf
4. https://github.com/Illutron/AdvancedTouchSensing
5. https://www.instructables.com/Touche-for-Arduino-Advanced-touch-sensing/
6. https://www.cnn.com/2023/03/30/world/plants-make-sounds-scn/index.html
7. https://www.riffusion.com/
8. https://medium.com/@vasily.onl/visualizing-sound-with-ai-e7a9191fea2c
9. https://github.com/Ajasra/Spout-for-Python
10. https://unitlondon.com/nfts/intelligence-all-the-way-down/
11. https://github.com/NVlabs/stylegan3
12. https://amarsaini.github.io/Epoching-Blog/jupyter/2020/08/10/Latent-Space-Exploration-with-StyleGAN2.html