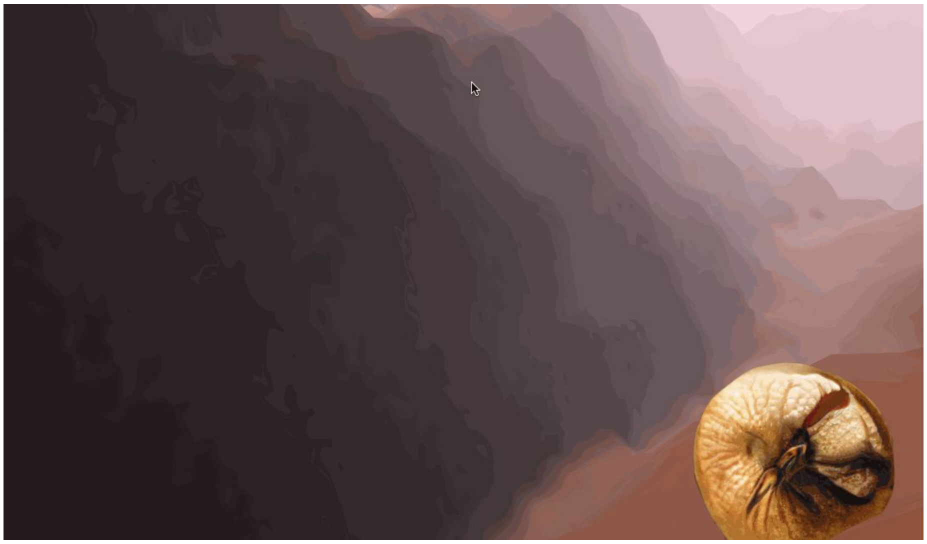
ART AND MACHINE LEARNING
CMU 2019 SPRING
PROJECT 2

Land of Misfit Creatures

Interactive at https://marisa.lu/experiments/landOfMisFits



Iris Hwang, Frank Kovacs, Subhodeep Mitra, Marisa Lu, Gillis Bernard

DESCRIPTION
Concept:
“The mule always appears to me a most surprising animal. That a hybrid should possess more reason, memory, obstinacy, social affection, powers of muscular endurance, and length of life, than either of its parents, seems to indicate that art has here outdone nature.” – Charles Darwin, *The Voyage of the Beagle*

During his study of the Galapagos Islands, naturalist Darwin reflected on how the mule, a cross between a male donkey and a female horse, inherits the best traits of both of its parents. For project 2, our team created hybrid animals and allowed machine learning–rather than nature–decide which physical characteristics were inherited. We “rigged the genetic lottery” throughout the process by adjusting values within Pix2Pix, DCGANS, and BigGANS to create the desired hybrid organisms. Ultimately, we created a digital ecosystem for our colorful, fantastical [and some nightmarish] creatures.

We were inspired by the following works:

- CycleGAN (1) – This paper explains how to use Pix2Pix and BigGANS show image-to-image translation without input-output pairs. While we didn’t ultimately employ this method, we admired the object transfiguration and the high resolution of the output image. In particular, the paper’s horse and zebra is a more refined, realistic version of our creatures.
- GAN Breeder (2)) – Ganbreeder.app is a collaborative online platform for ‘crossbreeding’ different image data sets (usually animals) to create interesting new images. They are essentially the GAN version of a very similar predecessor — Picbreeder, another collaborative art platform that proliferated “NEAT-based [NeuroEvolution of Augmenting Topologies] Genetic Art”. GANBreeder uses several models, but namely a 128x128 *BigGAN* image generator. The user interface allows non expert visitors to add data sets and toggle the weights of each as if changing the ‘genetics’ of their new creature.
- Animorphs (3) – During brainstorming, we discussed the memorable covers of the sci-fi young adult book series, Animorphs. The holographic covers are crude depictions the protagonist transforming into an animal. This quirky cultural relic spurred our imagination as we began to think of what animals we wanted to combine.
- Pokemon and Hayao Miyazaki characters – We also discussed the many forms that creatures take in the entertainment industry and their characteristics. Some appear to be similar to actual animals, such as Pokemon’s turtle-esque Squirtle, while others are fantastical, like Jigglypuff. Miyazaki’s Totoro is often seen as a hybrid of three different animals: the Japanese raccoon tanuki, a cat, and an owl.

Technique:
<u>Pix2Pix</u>

- First, we created our creatures through a number of different approaches. We drew via iPad, created outlines in Sketch with geometric shapes, and traced over photos of real animals. We also experimented with plants and non-living things as well, such as cars.
- We then found different datasets to add the hybrid element to our creatures. We used a public Pokemon dataset (4) and made our own dataset from a combination of a public Kaggle dataset of butterflies and found images via Google Search.
- We converted all of our pictures from PNGs to JPGS and ran Canny edge detection using openCV on our datasets. We then combined the original and outlined images, and split them into train and validation folders to prepare to train our model.

- We experimented with initializers, and adjusted (1) RandomNormal, (2) RandomUniform, and (3) a combination of RandomNormal and RandomUniform to observe how they impact the colors in the resulting creatures.
- We trained our model and ran the provided pix2pix notebook. In order to print our output, we added the code showX(fakeB) at the very last line and saved each image.
- Lastly, we placed our creatures within an interactive digital landscape built with JS. The online environment allows users to channel their inner explorers by allowing them to discover our novel creatures in a mysterious world.

<u>DCGANS</u>

- We experimented with DCGANs on different datasets, including the Stanford Dogs dataset (5) to create new “dogs” that have never been seen before. To get interesting images, we performed a hyperparameter sweep for the learning rate (0.0001 to 1), beta1 (0 to 1) and number of epochs (30 to 100).

<u>BigGANS</u>

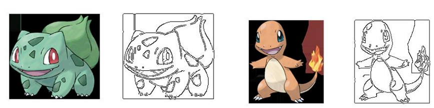
- We looked at the paper “Large Scale GAN Training for High Fidelity Natural Image Synthesis” by Andrew Brock, Jeff Donahue, and Karen Simonyan. For GANs of larger scale where the goal is generating high-resolution and diverse samples from ImageNet datasets, applying “orthogonal regularization” to the generator makes it amenable to a simple "truncation trick”. As such, there’s finer control over the trade-off between sample fidelity and variety by reducing the variance of the Generator's input. While we were inspired by GanBreeder, we stuck to interpolating between two datasets instead of trying to go through multiple.

Process:

<u>Pix2Pix</u>

Preparing outlines of creatures: First, we created files in Sketch of creatures we wanted to make based on our dataset. We created outlines in multiple ways: drawing via iPad, combining basic shapes, outlining pictures of real animals (e.g., a lion) with the pencil tool, and outlining a photo of a Pokemon character and adding other customized features.

Writing script for edge detection on image datasets: The paper used HED edge detection with post-processing to obtain the edge images. For our project, we decided to implement the Canny edge detection using openCV to get a simplistic edge images from our own datasets. After we obtained the edge image, we combined it with the source image to make it into the same format as required for the Pix2Pix algorithm. In order to obtain the best result for edge detection, we also created a script to convert PNG files to JPG. Instead of directly erasing the alpha channel, we filled the transparency with white pixels to avoid creating noises in the background.



Results with removing the alpha channel

Experimenting with RandomNormal and RandomUniform initializers: To see how the initializer for the tensor will affect the outcome, we tried the RandomNormal, RandomUniform, and a combination of the two initializers. From the results, the RandomNormal initializer gives a more robust result where some of the coloring is actually correct for some Pokemon. RandomUniform gives a more monochrome result where all the creatures have similar color outcome and gradient. The combination of Random Normal and Uniform provides a result that’s in between the results from RandomNormal and RandomUniform. The overall color and gradient is similar for all creatures but for some creatures, they have different color blocks.

<u>DCGANS</u>

A preprocessing step we decided to perform was image rescaling, akin to the flower dataset provided. This was done by batch running a script on all the dog images in the folder. Experimenting with the hyperparameters did not give vastly different results, but we did notice some slight trends. For example, learning rate affected the running time a lot (as we set the learning rate higher, the training got faster, but after a point around learning rate = 0.001, the image quality started deteriorating rapidly), and images became more refined after more epochs.

<u>BigGANS</u>

We varied seed noise and truncation values with a pretrained TensorFlow model and imageNet datasets in trying to find the best interpolated creatures. Our experimentation was methodical as we tried to isolate the effects of truncation vs noise, both in exploring a single dataset and when interpolation between two.

Varying truncation and noise value in a single dataset (pomegranate)

Varying truncation and noise value in while interpolating between 2 datasets

RESULTS

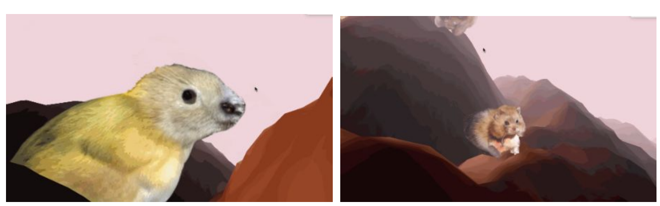
Pix2Pix

(Top to bottom: Outline, RandomNormal, RandomUniform, both RandomNormal and RandomUniform)

DCGANS

Attempt at creating new kinds of dogs with Stanford Dog dataset

BIGGANS



Hybrid creatures after isolating and balancing the effects of truncation vs noise in JS environment

CODE: https://github.com/deepm/artml/tree/master/proj2/p2p

LINK TO SUPPORTING MATERIALS WITH ALL PICTURES:
https://docs.google.com/document/d/1filJru35YW2CdMGIRvkBEvzQEXhu8BA4LGExXhNj0-4/edit

REFLECTION

Pix2Pix

- Ideally, we would have explored both the color blocking and the edge2photo options within Pix2Pix. It would have been interesting to see how our results differed if we had more control over where patterns were mapped on each creature.
- The result from training the Pix2Pix algorithm on 400 Pokemon images was impressively good; the output coloring almost matched the ground truth. We were surprised by how strong our results were.
- Running the same model through everyday objects and animals created interesting results. The results were colorful, but far from the ground truth. We believe this was because the form of everyday objects and animals were very different from the Pokemon (i.e. lines are more complicated, and the shape and thus the silhouette of the subject are different).

- We also experienced on the effect of initializers on the final result. From the results, RandomNormal gives the results that most closely resemble the ground truth; RandomUniform creates monochrome results; and the combination of RandomUniform and RandomNormal creates outputs that are similar to the results of RandomNormal, but with a darker tone/shade on the characters.
- After running our own outline images through these initializers, we found the RandomNormal and the combination of RandomNormal and RandomUniform gives the best results. The result images are very colorful and add a fantastic element to the creatures, resembling an animal that you might see in a child's book.
- In the future, we would like to try more different initializers and their parameters for each layers and see how they may affect the final results.

Results after running 400 images to train the model
(Top:Edge, Middle: Ground Truth, Bottom: Output)

Changed Initializers

Results using both RandomNormal and RandomUniform as initializers

<u>DCGANS</u>

- The final images of dogs visualized weren't as impressive as we had hoped. They were quite blurry despite experimenting with multiple different hyperparameters (learning rate, beta1, number of epochs).

<u>BigGANS</u>

- The best results seemed to yield from datasets with very consistent standard backgrounds — for example, the hummingbird dataset always had blurred solid color backgrounds that didn't interfere with the interpolation process (a side effect of how hummingbirds are photographed). Best results yielded from simple backgrounded datasets. A dataset like goldfish always had refraction issues from the water or colorful blocks in the back, likely as a result of how goldfish are photographed in tanks. Taking in this additional meta information about how the dataset might be captured played a great role in how to curate the interpolated breeding.

REFERENCES

1. CycleGan (https://github.com/junyanz/CycleGAN, https://arxiv.org/pdf/1703.10593.pdf)
2. GANBreeder (https://ganbreeder.app/)
3. "Large Scale GAN Training for High Fidelity Natural Image Synthesis" (https://arxiv.org/abs/1809.11096)
4. Animorphs (https://en.wikipedia.org/wiki/Animorphs)
5. Stanford Dogs dataset (http://vision.stanford.edu/aditya86/ImageNetDogs)
6. Pokemon.com Dataset (https://www.pokemon.com/us/pokedex/)

ROLES & RESPONSIBILITIES:

Our team took an apprentice approach to learning. Each person contributed specific skills and shared their knowledge with the rest of the team. This method empowered each of us to complete the project deliverables and explore our concept space with confidence.

- Iris Hwang: Outline and file type conversion scripts, auxiliary scripts to help create training data and inputs, initialization exploration
- Subhodeep Mitra: Experiments with DCGANs on the Stanford dogs dataset
- Frank Kovacs: Report proofreading, exploration of BigGANS
- Marisa Lu: JS environment creation, exploration of BigGANS
- Gillis Bernard: Report composition and project management