Dr. Badri Vellambi
vellambn@ucmail.uc.edu

Project Advisor

Tim Neale
nealetw@mail.uc.edu

Daniel Hackney
hacknedn@mail.uc.edu

Team: ShenaniGANs

Building Generative Adversarial Networks

Project Purposes:

- Develop group members' knowledge of neural networks and AI concepts
- Design a reliable and practical image generator

Goal Statement:

Develop a functional generative adversarial network, capable of producing convincing images of real-life phenomena.

Goals and Background

- Design a functioning GAN that can generate images from input data
- Learn more about the technology and mathematics behind GANs
- Experiment with GAN optimization and manipulation
- Design a demonstration of GANs showing their utility and versatility

The background of this project stems from an interest in Machine Learning as well as online examples of what kind of image generation GANs are capable of, such as what can be found on www.thispersondoesnotexist.com. We wanted to delve deep into how GAN software is constructed, what can improve them, and how modifying various aspects of their inputs impacts their output.

Intellectual Merits

With our personal interest in GANs, the merit behind our project is for the sake of personal learning, as we wanted to see just how this sect of computer science and data analysis has evolved into something more useful and interesting.

The merits behind it are a bit novel as well, with how new the technology is. Learning how this new technology is used and where it will be going is useful for anyone in computer science. We figure that putting more open-source code relating to GANs out in the wild will be helpful for future endeavors into this area.

Broader Impacts

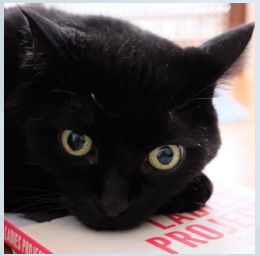
GANs will have a large impact in the future of both generating original digital works, as well as classifications for objects in digital images. Beyond simple text and image generation, future GANs may have the capability to produce entire novels, movies, or songs based on existing versions of such things. Exploring and optimizing GANs is crucial in achieving these lofty goals.

Our experience will help us further the field of generative networks, and in the future we would know how to optimize and improve networks that classify and generate images of everyday objects.

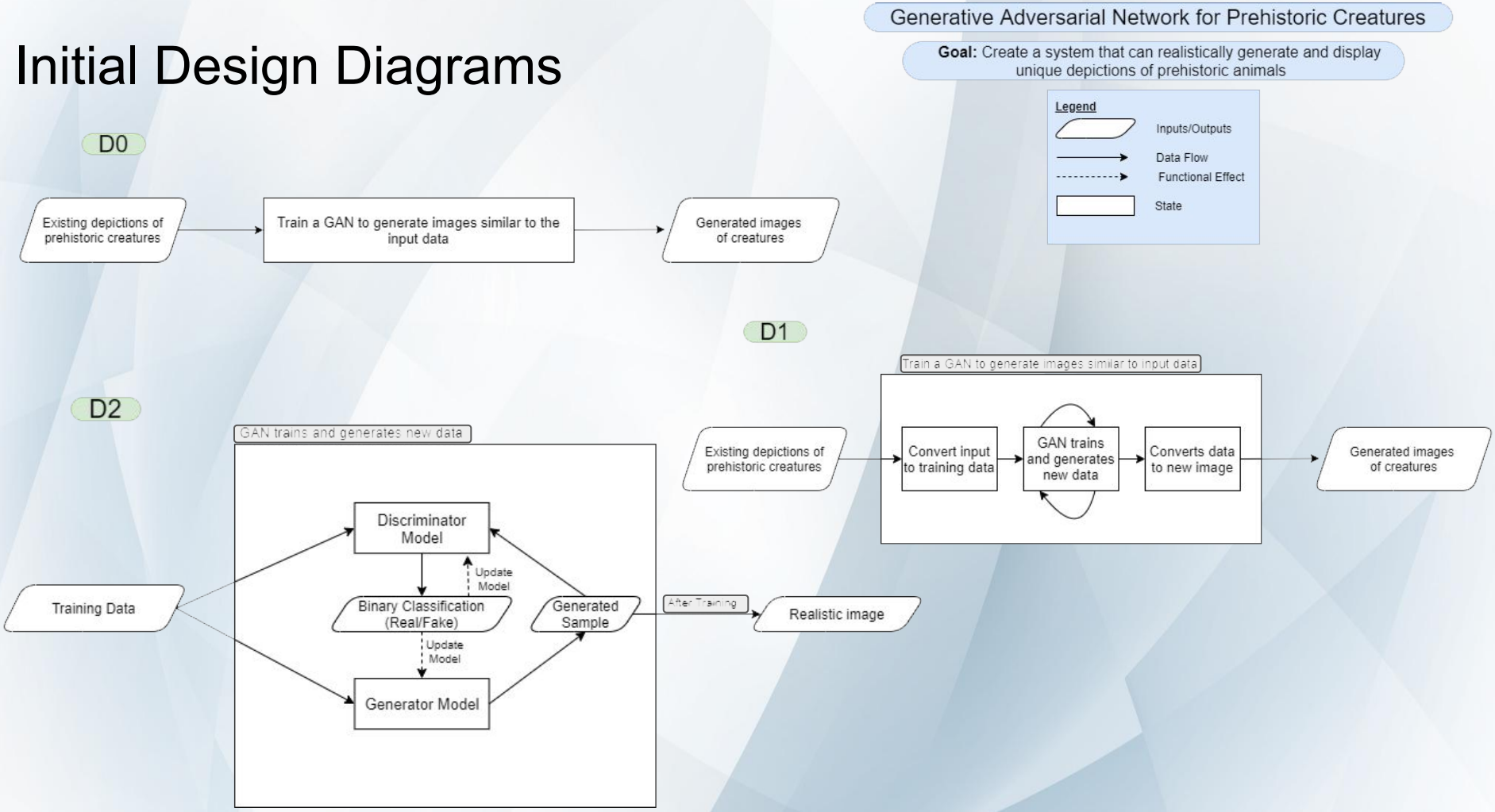
Design Specifications

Our GAN, like most, is composed of a generator and discriminator model. Although we initially wanted our input domain to consist of dinosaurs, we were unable to find a large enough database of consistent renditions. We have since decided to focus on images of house cats, as large databases are readily available for this domain.

The generator receives images of cats, attempts to devise one of its own, and then receives correction from the discriminator. As training continues, the generator will generate more realistic images, while the discriminator gets better at determining which ones are generated.



Initial Design Diagrams



Technologies Used

The basis of our GAN is Python using TensorFlow, which is a massive library for working with neural networks. We use several functions from matplotlib, numpy, and the Keras API as well.

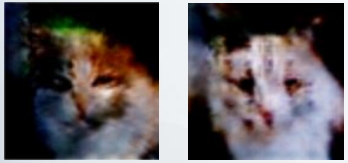
While researching, we also ran into things like PyTorch for more neural networks and OpenCV for image processing.

We also used parallel/cloud computing, as we ran some of our network training on the Ohio Supercomputer (OSC).

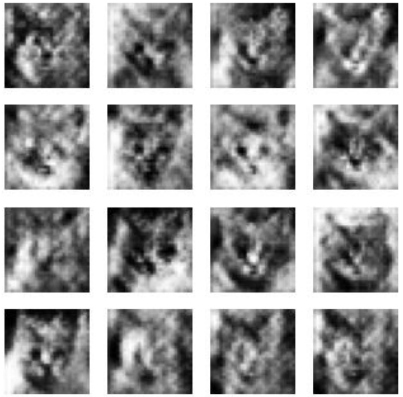
Milestones

Task	Time spent	Results
Preliminary Research	Fall and much of Spring	Basis knowledge of how GANs work, GAN math
28x28 Black and White Cats	4 weeks	A modified MNIST GAN to generate tiny cat faces rather than digits
128x128 Cats	2 week	GAN to generate larger cat faces than before
Colored Cats + Info Graphs	4 weeks	Final output for our GAN, with 128x128 colored cat faces

Results/Deliverables

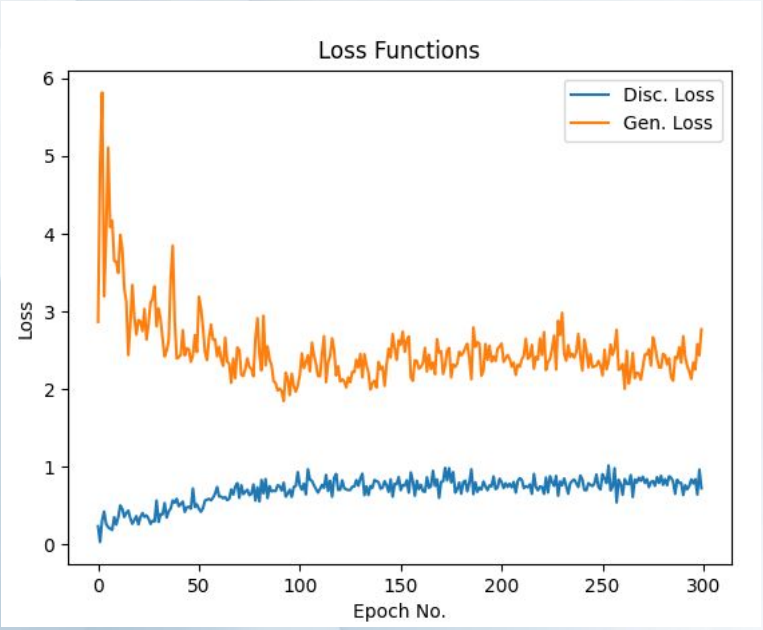


A couple of our personal favorite cats generated by our GAN



Initial 28x28 pixel black&white cat faces

Final 128x128 pixel RGB cat faces



The loss functions of our generator and descriminator models for a sample run of 300 epochs

Challenges

Scope: After researching, we thought implementation of a GAN would be much easier than it actually is. From the beginning, we knew it would be difficult, but over time, we had adjusted our goals to reflect what we could get done with the resources we had access to.

Computational power and time: Running our neural network on our home computers was taking a lot of time, and we didn't see results from changes we had made for several hours. This was somewhat alleviated with the use of our GPUs, but training time still significantly impacted our development time.

Database Availability: There was much less variability and availability of large enough datasets for GANs to run on than we expected. Our initial plan to generate dinosaurs was then swapped for cats instead.