GANimals

CS 455 Final Project

Daniel Khalil, Luke Crump,
Vivian Dang

Overview of Challenge

- There was a need for mass creation of monstrous animal faces.
- Doing this traditionally, either by hand-drawing or photoshop, proved to be too time-consuming.
- Thus, GANimals was born.

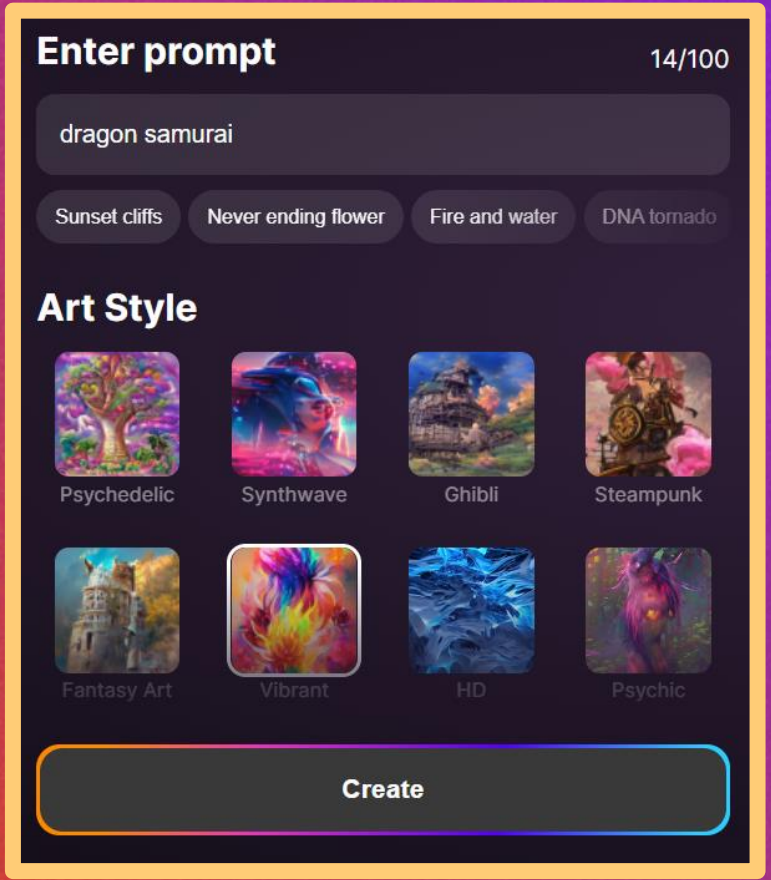
Limitations

- Hardware
  - Most of our computers were not strong enough to handle generating the images.
- Dataset
  - Kaggle dataset is limited to mammal faces only, but “GANmals” wasn’t as catchy.

Approach

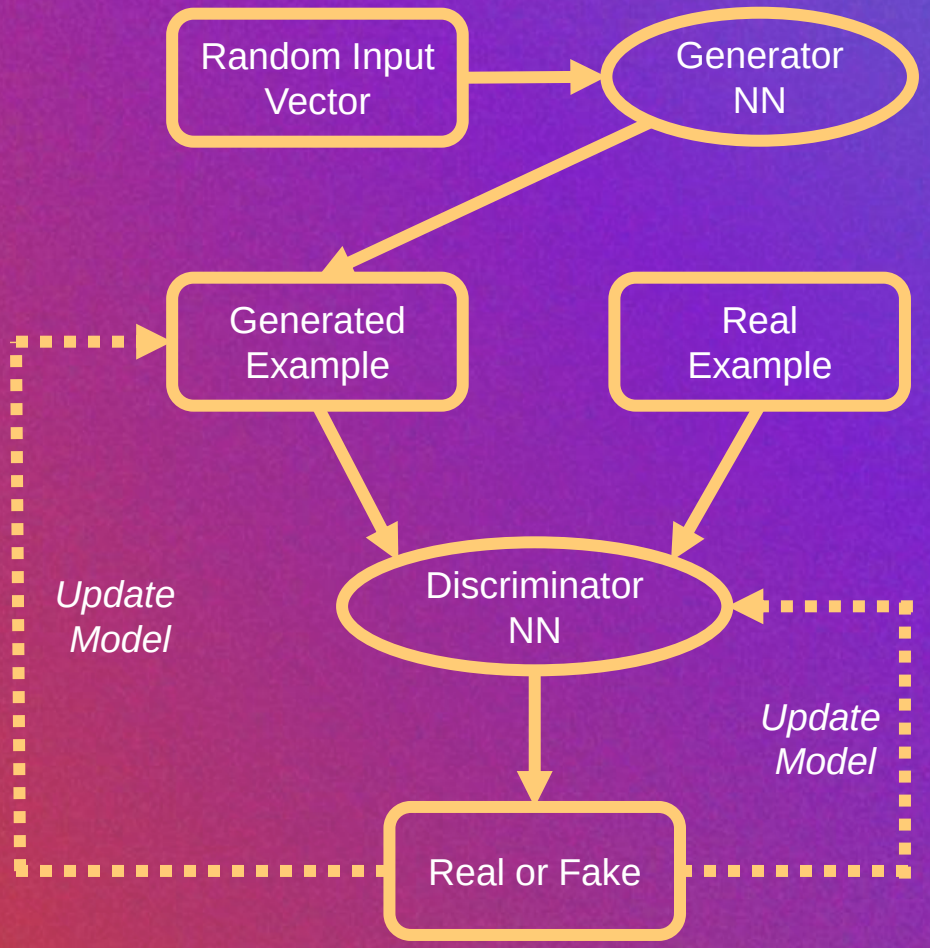
- The images were generated using GANs, otherwise known as Generative Adversarial Networks.
- There have been many other applications in art* that also utilize GANs.

*Ours is not this good.



Algorithm

- Two neural networks:
  - Generator – creates fake sample
  - Discriminator – attempts to discern whether or not it is “real”
- They are trained to compete against each other.

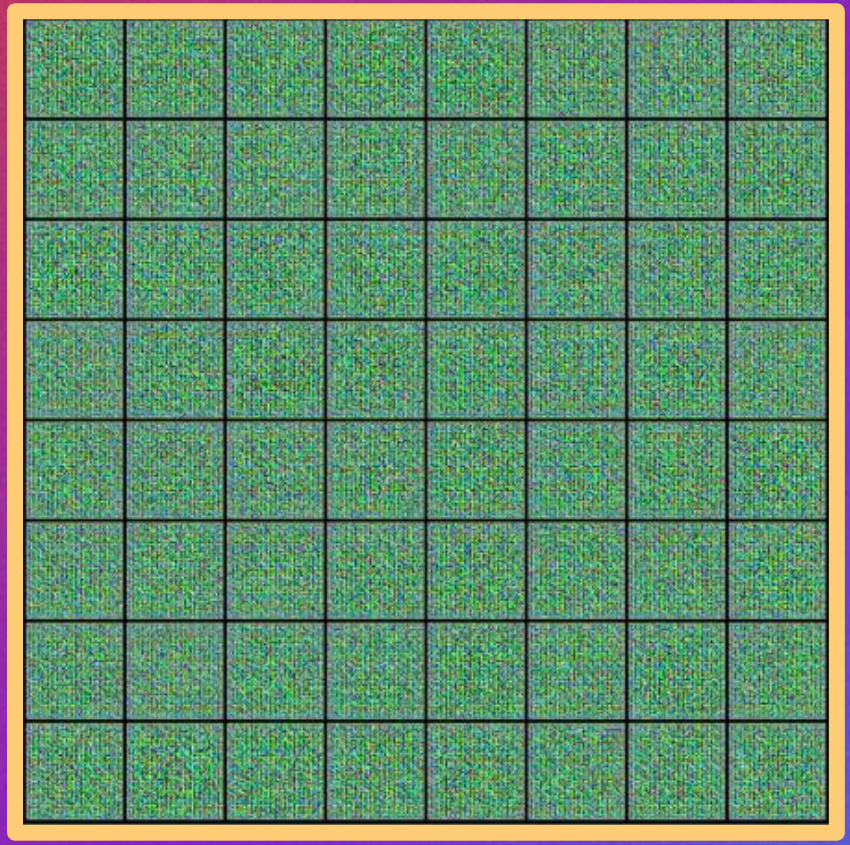


Key Resources

- Kaggle Animal Faces Dataset which includes 16,130 high-quality images at 512×512 resolution.
- The images were split into three categories:
  - Cats
  - Dogs
  - Wildlife

Solution

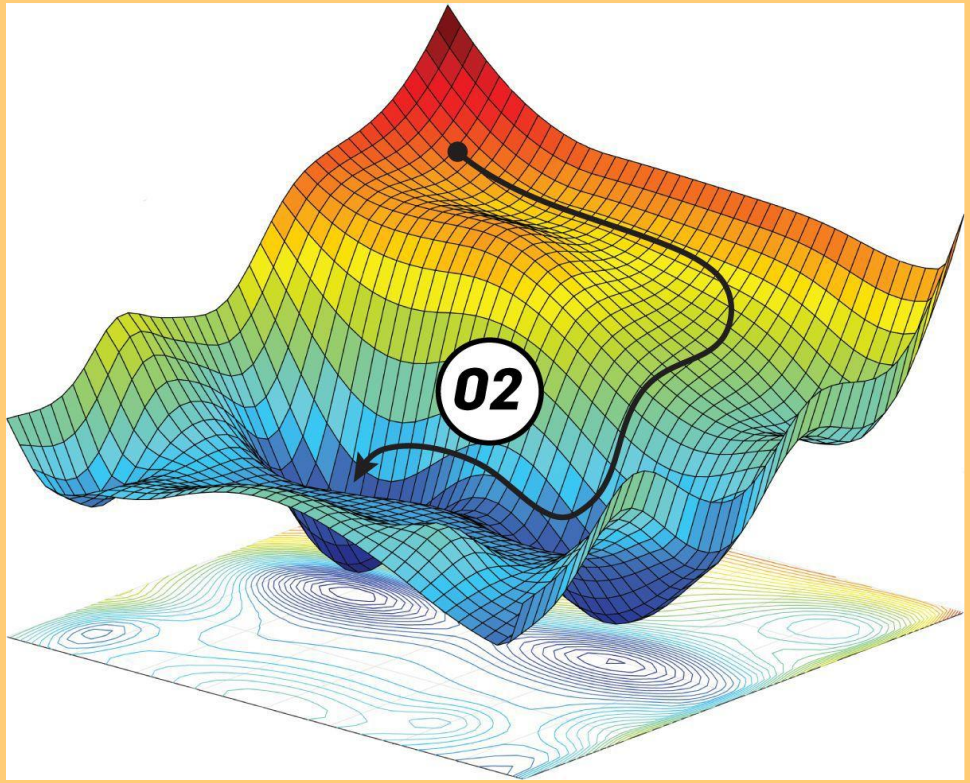
- GENERATOR MODEL
  - Input is a matrix of random numbers (Latent Tensor)
  - Used as a seed for generating “fake” images
  - Convert Latent Tensor into an image using Transposed Convolution (a.k.a. Deconvolution) which is shown to the right



Solution

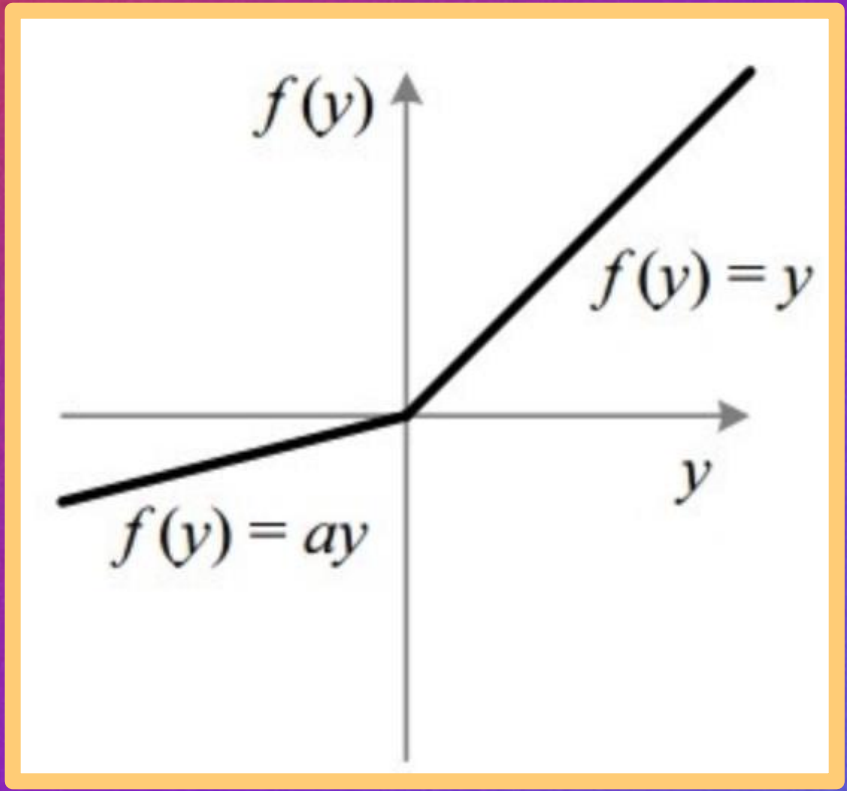
GENERATOR TRAINING

1) Generate a batch of images and then pass it to the discriminator.
2) Calculate the loss by setting the discriminator's target labels to 1. (We're trying to trick the discriminator.)
3) Use resulting loss to perform Gradient Descent to update the weights of the generator.



Solution

- DISCRIMINATOR MODEL
  - Convolutional Neural Network
  - Outputs a single number per image between 0 and 1
  - Stride = 2
  - Input = 64 x 64 (Dataset was scaled down)
  - Activation Function: Leaky Rectified Linear Unit (Leaky ReLU) shown to right



Solution

- DISCRIMINATOR TRAINING
  - Loss Function = Binary Cross-Entropy

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Binary Cross-Entropy / Log Loss

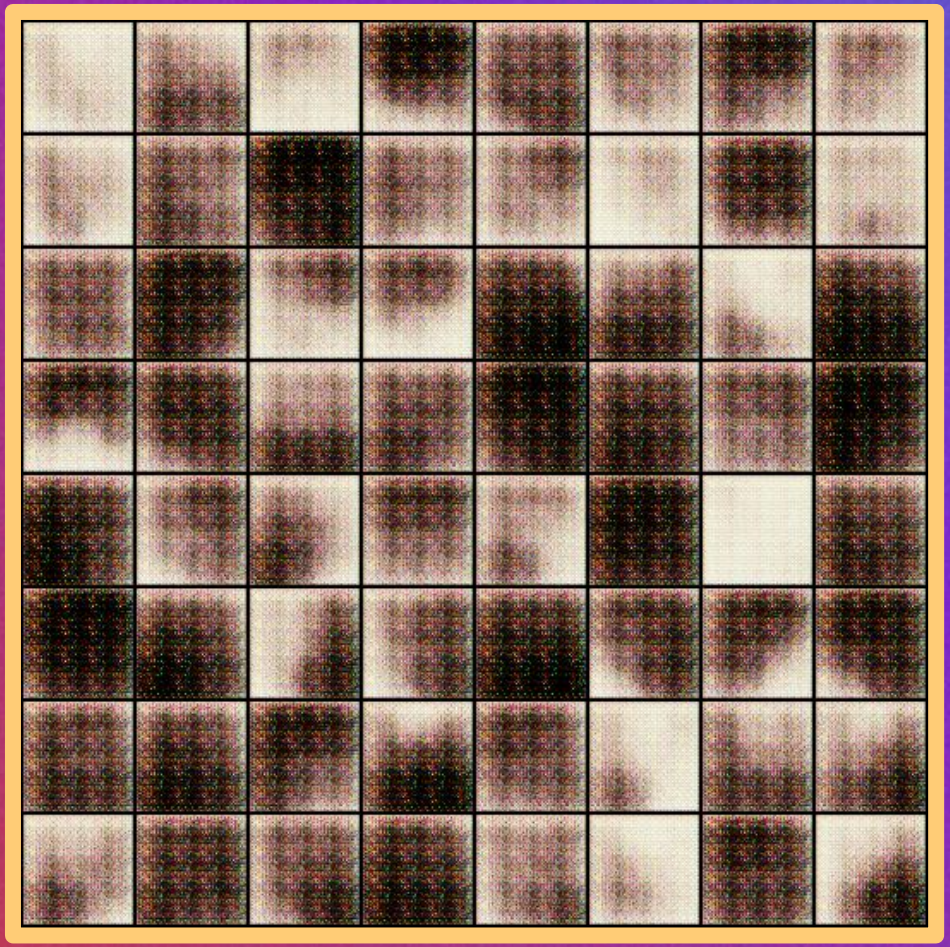
1) Pass a batch of real images, compute loss, and then set target labels to 1.
2) Pass a batch of generated images, compute loss, and then set target labels to 0.
3) Add the two losses and use overall loss to perform Gradient Descent to update the weights of the discriminator.

Demonstration

- Due to limitations of hardware, a live demo was not viable.
- However, we were able to generate up to 50 epochs before danger was imminent.
- Please prepare yourselves for the following images.

Demonstration

- This is the first epoch that we generated.
- As you can see, there's not much to discern yet.



Demonstration

- At 25 epochs, we're really seeing some movement.

Demonstration

- 50 epochs in, and there isn't really a noticeable difference between 25 epochs and this.

Demonstration

Demonstration

Conclusion

- Overall, it seems like our call for the creation of monstrous animal faces was answered.
- Operation GANimals was a success.

Thank You!

Any Questions?