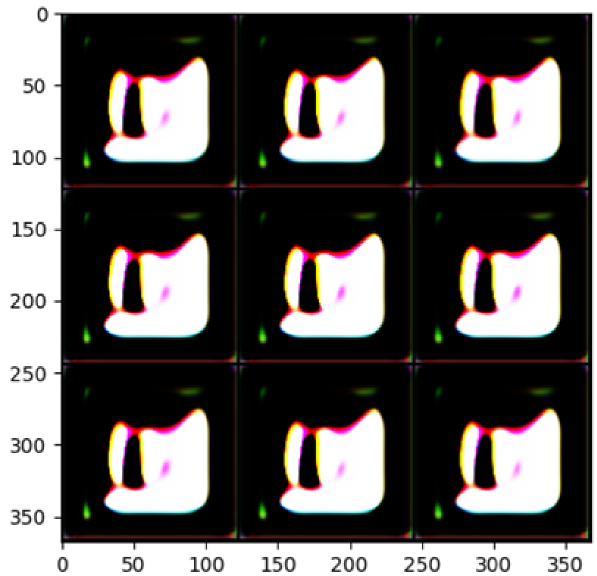

Pokemon Generator

10-615 2022 Spring Project 2

Erica Weng Emma Kwan Zhiping Liu Lanxuan Zhou

1 Group Information

1.1 Group Number: 6

1.2 Group Members

- Lanxuan Zhou: Master of Information Systems Management (MISM), Heinz College
- Zhiping Liu: Master of Information Systems Management (MISM), Heinz College
- Erica Weng: Robotics Machine Learning, CMU Robotics Institute
- Emma Kwan: Information Systems, Dietrich College of Humanities and Social Sciences

2 Description

2.1 Concept

With the development of various conditional generative adversarial networks (Conditional-GAN), generating images based on all kinds of inputs become possible. As long as you design a well-structured networks and have enough time to train your model, you will have a generator by which you turn something you want into an image.

Pokemon is a video game series with 26-year history. In the game, there are creatures called Pokemon. Based on their types, abilities, stats, and habitat environments, they tend to have different images. The motivation of our project is to let people design their own Pokemon by indicating Pokemon's attributes via Conditional-GAN.

2.2 Technique

2.2.1 Generative Adversarial Network

GAN is a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G [**1**]. The reason why it is called adversarial is that there are two networks competing with each other. During the training, the generator network is trying to fool the discriminator network, and the discriminator is trying to distinguish generated images from real images.

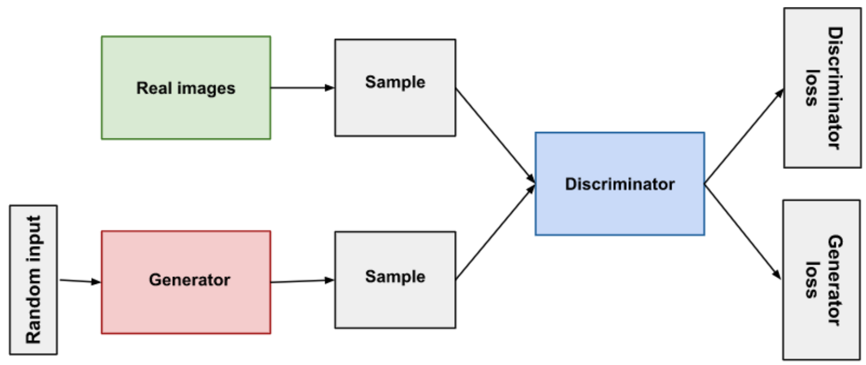

Figure 1: Structure of GAN

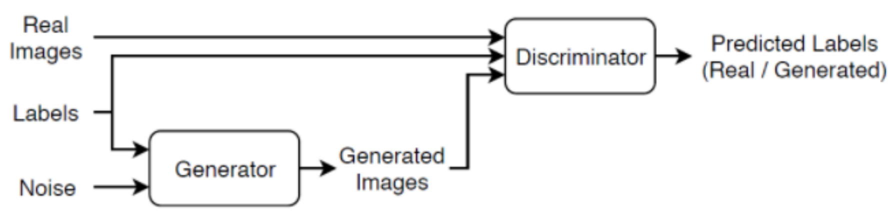

Figure 2: Structure of Conditional-GAN

2.2.2 Conditional-GAN

By training GAN, we will obtain a generator which can generate images with random noise as the inputs. However, what if we want to input something to get different images? Here we have Conditional-GAN. It is the conditional version of generative adversarial nets, which can be constructed by simply feeding the data, y, we wish to condition on to both the generator and discriminator[**2**]. By training with certain labels added to generator and discriminator, we can control the output of generators with user inputs.

2.3 Process

We used the architecture provided on this website[**3**] for building a Conditional GAN. However, this model used an MLP for both the discriminator as well as the generator. The MLP architecture does not provide the necessary structure to generate diverse, detailed images, so we were not able to get very good results.

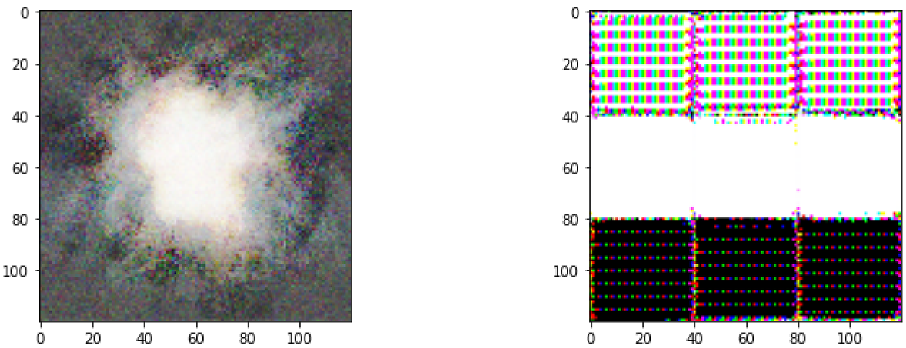

Figure 3: Initial Outputs

We referenced this website[**4**] as a reference for designing a convolutional architecture for the generator as well as the discriminator. The Generator is composed of a series of deconvolutional layers which upsample the image to create a larger image with each layer, and the discriminator composed of a series of convolutional layers with downsample the input until we receive a linear embedding, from which the discriminator can make a classification decision.

2.4 Reflection

We have some of the ideas for next steps. First, we can try a smaller image size. With a smaller image size, the model has to create less details, and thus may be able to learn generation. Second, we can try to label Pokemon with not only their primary types, as we have already done, but also their secondary types. This way, each pokemon is matched to multiple types, and has multiple "correct labels." In addition to processing the additional labels, we would also have to create a custom loss function to accept multiple correct target values for certain pokemon images. This may help our model learn better, as there will be more pokemon images per type, and thus the model will have more examples of visual features to learn from for a certain type. Third, we can try to add different images of the same Pokemon. We found datasets with multiple images of the same pokemon, pictures taken from different angles of the pokemon in different positions. These may be good augmentations to the dataset, and provide more data for learning. Fourth, we can try to augment the dataset ourselves with translations, rotations, and other transformations of the dataset. More data may help the model learn quicker.

3 Result

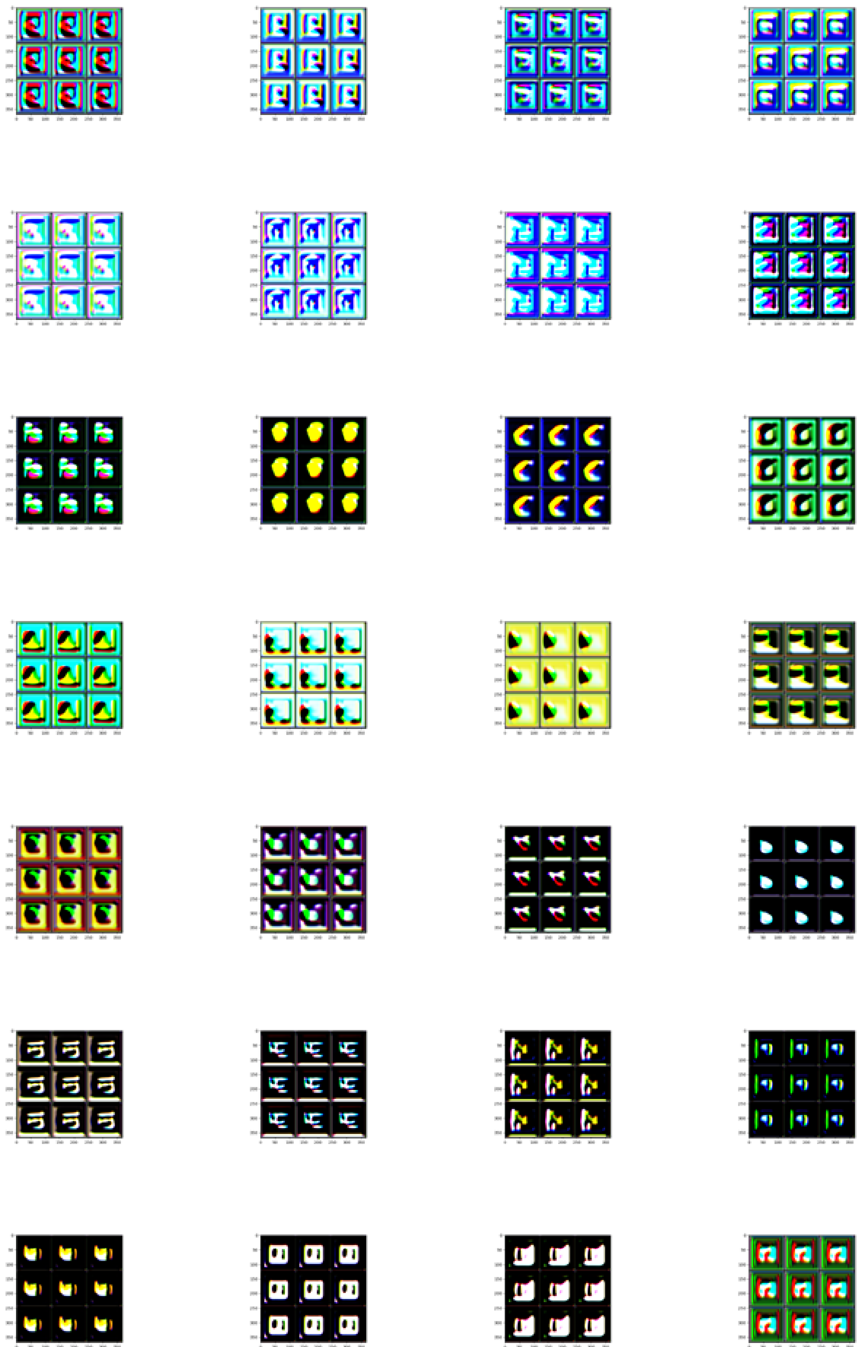

Figure 11: Outputs with different user inputs

4 Code

Our codes are on Google Drive

5 Reference

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozairy, Aaron Courville, Yoshua Bengio *Generative Adversarial Nets*

[2] Mehdi Mirza, Simon Osindero *Conditional Generative Adversarial Nets*

[3] Conditional GAN (cGAN) in PyTorch and TensorFlow

[4] Github togheppi/cDCGAN