Generating Pokémon Using Stable Diffusion
CS680 Project Proposal

Ali ElSaid
Department of Statistics and Actuarial Science
University of Waterloo
ali.elsaid@uwaterloo.ca

1 Introduction

Pokémon is the single highest grossing media franchise in history worldwide, accumulating over $ 118.5 billion in revenue since its year of inception, 1996, across video games, anime, movies, merchandise and more. As of today, there are 905 unique Pokémon (short for Pocket Monsters) and tremendous effort is spent into designing each and every one of them. Wouldn't it be great if we were able to generate completely new Pokémon with just the click of a button or a simple short description? This is where the currently widely trending generative deep learning models come in. The goal of this project is to use the Stable Diffusion model to generate new Fake-mons (Fake Pokémon) from short text prompts that are as high quality as possible and that can be mistaken for deliberately designed real Pokémon. I am a huge Pokémon fan and I grew up playing and watching it, and I still do until now. So, the thought of creating my own new Pokémon in a matter of minutes excites me. This will also be my first experience with deep learning, so I am using this project as a gateway into deeply learning the world of deep learning.
Aside from my personal goals, this project could be eventually made in the future into a website that provides an interface for Pokémon fans (especially children) that enables them to create their very own Pokémon. These newly generated Pokémon could also serve as a seed for new design ideas for the creators of Pokémon themselves, easing the design process and enabling the generation of novel designs and combinations they would have never thought of. Also, the model could be used to generate beautiful AI generated artwork for people to enjoy and use as their backgrounds or other similar usages.

2 Related Works

As far as academic papers go, there are not many that have explored the topic of Pokémon generation. I have seen two papers that are related to Pokémon and deep learning. The first [1] is a paper that attempts to generate new fake Pokémon trading cards (for the Pokémon trading card game (PTCG)) using deep learning. The authors used Generative Adversarial Networks (GANs) and Conditional Adversarial Networks to generate the images, and they used Char-Recurrent Neural Networks (RNNs) to generate the text for the cards; their results were mediocre. The second paper [2] applies style transfer techniques using Convolutional Neural Networks (CNNs) and GANs (CartoonGAN, CycleGAN and StyleGAN) to transform the style of photos of pets to be in the style of Pokémon; the results of this paper weren't great. The main contributions I found for Pokémon generation using deep learning were from articles on the internet. Most of them trained variations of GANs from scratch to generate new Pokémon images, however most of their performance was not satisfactory; the Pokémon images were too blurry and had no distinct features (References [3] to [9]) see Figure 1.

Based on Professor Yu's recommendation I began looking into Diffusion models (specifically Stable Diffusion), I found an article [3] that fine tunes the Stable Diffusion model on Pokémon images to have it generate Pokémon-style images from prompts (for example, if the prompt is "Donald

Trump", the model will generate an image of Donald Trump in the form of a Pokémon, see Figure 2).

This was the closest attempt I found to what I aim to do, and the most successful one too. I also found a bunch of articles that are unrelated to Pokémon, but describe how to fine tune the Stable Diffusion model using multiple methods, one of which is called Textual Inversion [4] which I will be using as references to create my own version of fine tuning Stable Diffusion.

Figure 1: Results of using GANs to generate Pokémon.

3 Plan

I will start with studying the generative models, namely GANs and diffusion models. I will then research the topic of prompt engineering and will attempt to get the best possible results I can out of the base Stable Diffusion model by providing the best prompts (some initial results of experimenting with prompts on Stable Diffusion can be seen in Figures 3 and 4.

I will then reproduce the results of the above-mentioned article [3] to get familiar with how the model works and how to practically use it and fine tune it. I will then experiment with other methods of fine tuning (namely Textual Inversion), different configurations and different Pokémon datasets have the model generate the highest quality Pokémon I can acheive.
My focus will not only be on generating totally new Pokémon, but also already known Pokémon in new different scenarios based on text prompts; for example, a prompt could be "Bulbasaur eating an apple" and I will attempt to fine tune the model to generate a high-quality image that matches that prompt. Finally, I will do some more prompt engineering to figure out the best way to provide prompts to get the highest quality results.

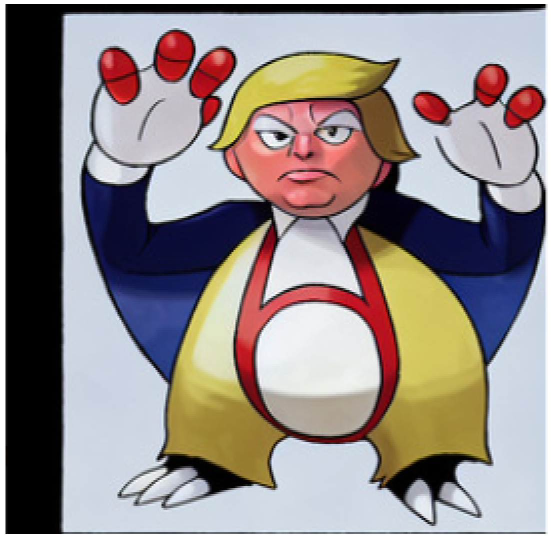
Figure 2: Donald Trump as a Pokémon.


Figure 3: Prompt: ”Bulbasaur”.

Figure 4: Prompt: ”Greninja”.

The performance of the model will be evaluated by comparing the output Pokémon images to the already existing official Pokémon and to the output of the articles used as references by eye as there is no numerical way of judging the performance of the model. The characteristics I will be using to judge the performance of the model include but are not limited to shape, quality, clarity, distinct features, cohesiveness of look and colors, appeal as a potential Pokémon design etc.

As for the data, there are numerous sources of data available. [5] is a huge database of images for all 905 Pokémon in all forms (Pixel sprites, hand drawn images, 3D models, official artwork, etc.) which could be very useful if I would like to train the model to generate a specific type of Pokémon images, or if I would like to train it on all kinds of them so it would be flexible in generating any of them. This database also has images for all the “shiny” versions Pokémon, which are basically the same Pokémon but with an altered color. This could prove useful in augmenting my data in the case that the base 905+ images are not enough for my application. Also, in that case, there are some other initial ideas for augmenting my data, which include: mirroring the images, adding random noise to the images and recoloring the images. Another idea, that will however only work in the case of training on the pixel sprite image data, is using online automatic generators for “fusions” of Pokémon sprites. Which are basically websites that take two Pokémon and fuse their pixel sprites to output an imaginary Pokémon resulting from the fusion of the two original ones, which can be considered as a new Pokémon and used to augment my data. There are also other sources of data listed in the references that could potentially be used instead of or in conjunction with the mentioned database. A path that might also be explored is using the Pokémon trading card game data.

As for the resources needed, from the articles I read, it seems like fine tuning the model can be managed with one GPU that has a VRAM of at least 24 GB. I plan to either use university resources for that if possible, or if that is not possible I will purchase GPU hours from the Lambda GPU Cloud to train the model as I am personally excited about this project and invested in it.

References

[1] Michael B. Hedge, Morgan Nelson, Thomas Pengilly, and Michael Weatherford. "PokéGAN: P2P (Pet to Pokémon) Stylizer". *SMU Data Science Review*. 10th ser., vol. 5, no. 2 (2021) (cit. on p. 1).

[2] Devi Acharya, Rehaf Aljammaz, Beth Oliver, and Mirek Stolee. *Gotta Generate 'em All! Pokemon (With Deep Learning)*. 2019 (cit. on p. 1).

[3] Justin Pinkney. *How to fine tune stable diffusion: How we made the text-to-pokemon model at lambda*. Oct. 2022 (cit. on pp. 1, 2).

[4] Foong. *How to fine-tune stable diffusion using textual inversion*. 2022 (cit. on p. 2).

[5] PokeAPI. *Sprites/sprites/pokemon at master pokeapi/sprites* (cit. on p. 4).

[6] Justin Kleiber. *Pokegan: Generating fake pokemon with a generative adversarial network ...* June 2020.

[7] Tyler Folkman. *How To Create Unique Pokémon Using GANs — by Tyler Folkman — Towards ...* 2020.

[8] Digvijay Yadav. *Pokemon Generator Gans*. July 2020.

[9] theRoughCode. *TheRoughCode/pokegan: Pokemon generator using the power of gans.*

[10] Arham Lodha. *My journey through gans for pokemon generation*. July 2020.

[11] Menghanibhanvi. *Pokemon Generation Using Gans (pytorch)*. Nov. 2020.

[12] Mehta. *Generating New Pokemons using Gans — by Jatin Mehta — becoming ... - medium*.

[13] Rayfield. *AI generated pokemon sprites with GPT-2*. 2020.

[14] LambdaLabs. *Lambdalabs/SD-image-variations-diffusers · hugging face*.

[15] ExponentialCookie. *R/stablediffusion - [tutorial] "fine tuning" stable diffusion using only 5 images using textual inversion.*

[16] zoru22. *R/stablediffusion - I got stable diffusion to generate competent-ish leavannies w/ textual inversion!*

[17] External_Quarter. *R/stablediffusion - let's discuss best practices for finetuning.*

[18] P.w. *Fine tuning stable diffusion images with cross attention control*. Sept. 2022.

[19] Joy Zhang. *I spent $15 in dall·e 2 credits creating this AI image, and ... - medium*. 2022.

[20] Kvpratama. *Pokemon Images Dataset*. Aug. 2020.

[21] Lambda. *Lambdalabs/pokemon-blip-captions · datasets at hugging face*.

[22] llSourcell. *Pokemon GAN/data at master llSourcell/pokemon gan*.

[23] *Pokémon sprites: Archive of pokémon images from every game*.

[24] *Sprite packs*.