CS680 Project Proposal

Generating Pokémon Using Stable Diffusion

Introduction

Pokémon is the single highest grossing media franchise 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. As for myself, 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’ve 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.

Pokémon is the single highest grossing media franchise 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 there are a bunch more coming out very soon with the release of the new games Pokémon Scarlet and Violet on 18th of November 2022. Tremendous effort is spent into designing these Pokémon and every single one of them has a unique design and combination of elemental type, stats, egg group, attack moves it can learn, abilities and so much more. Pokémon is one of my favorite things ever as I quite literally grew up playing and watching it. And to this day, I still play every Pokémon game that comes out yearly. So, the thought of creating my own Pokémon is a very exciting one for me. And since I am not the most artistic of people, that dream would normally be impossible for me. However, that is where

Initial Literature Review

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]). Based on Professor Yu’s recommendation I began looking into Diffusion models (specifically Stable Diffusion), I found an article [11] that fine tunes the Stable Diffusion model on Pokémon images to make 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. 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 that describe how to fine tune the Stable Diffusion model using multiple methods, one of which is called Textual Inversion [10].

Plan

I will start with studying the generative models, namely GANs and diffusion models. I will then research the area of prompt engineering and will attempt to get the best possible results I can out of the base Stable Diffusion model. I will then reproduce the results of the above-mentioned article [11] to get familiar with how the model works and how to practically use it and fine tune it. I will then attempt to experiment with other methods of fine tuning (namely textual inversion), different configurations and different data to get the model to generate new high quality Pokémon. My focus will not only be on generating totally new Pokémon but also already known Pokémon but 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. I will then do some more prompt engineering to figure out the best way to provide prompts to model to get the highest quality results. 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. And lastly, if there is time, I will try to caption my own dataset and fine tune the model on it to enable the generation of Pokémon from prompts based on specific attributes like type (e.g. fire, water, grass, steel, etc.), height, weight or egg group (e.g. human-like, monster, bug, flying, dragon, etc.).

As for the data, there are numerous sources of data available. [18] is a huge database of images for all 800+ 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 a different color. This could prove useful in augmenting my data in the case that the base 800+ 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 using 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 could also be explored could be 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 very excited about this project and invested in it.

[1] https://creativecoding.soe.ucsc.edu/courses/cmpm202\_w19/projects/GottaGenerateEmAll\_ACM\_Project3.pdf

[2] https://scholar.smu.edu/cgi/viewcontent.cgi?article=1193&context=datasciencereview

Pokemon GANs:

[3] https://blog.jovian.ai/pokegan-generating-fake-pokemon-with-a-generative-adversarial-network-f540db81548d

[4] https://towardsdatascience.com/how-to-create-unique-pok%C3%A9mon-using-gans-ea1cb6b6a5c2

[5] https://www.kaggle.com/code/digvijayyadav/pokemon-generator-gans

[6] https://github.com/theRoughCode/PokeGAN

[7] https://medium.com/@arhamsmita/my-journey-through-gans-for-pokemon-generation-c8662de64a1d

[8] https://menghanibhanvi.medium.com/pokemon-generation-using-gans-pytorch-ede11baf4069

[9] <https://becominghuman.ai/generating-new-pokemons-using-gans-ceba1c6dc676>

<https://matthewrayfield.com/articles/ai-generated-pokemon-sprites-with-gpt-2/>

Pokemon Stable Diffusion:

Fine Tuning Stable Diffusion:

[10] https://towardsdatascience.com/how-to-fine-tune-stable-diffusion-using-textual-inversion-b995d7ecc095

[11] https://lambdalabs.com/blog/how-to-fine-tune-stable-diffusion-how-we-made-the-text-to-pokemon-model-at-lambda/

[12] https://huggingface.co/lambdalabs/sd-image-variations-diffusers

[13] https://www.reddit.com/r/StableDiffusion/comments/wvzr7s/tutorial\_fine\_tuning\_stable\_diffusion\_using\_only/

[14] https://www.reddit.com/r/StableDiffusion/comments/wz88lg/i\_got\_stable\_diffusion\_to\_generate\_competentish/

[15] <https://www.reddit.com/r/StableDiffusion/comments/wywxgs/lets_discuss_best_practices_for_finetuning/>

<https://reticulated.net/dailyai/fine-tuning-stable-diffusion-images-with-cross-attention-control/>

https://pub.towardsai.net/i-spent-15-in-dall-e-2-credits-creating-this-ai-image-and-heres-what-i-learned-52f352912025

Data:

[16] https://veekun.com/dex/downloads

[17] https://pokemondb.net/sprites

[18] https://github.com/PokeAPI/sprites/tree/master/sprites/pokemon

[19] https://www.kaggle.com/datasets/kvpratama/pokemon-images-dataset

[20] https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions

[21] https://github.com/llSourcell/Pokemon\_GAN/tree/master/data