Diffusion Model for Generating new Pokémon

Paper

Michael Hüppe

08.12.2023

Abstract

Making its gaming debut in 1996 with "Pokémon Red and Green" and earning approximately 368 million from game sales alone, the Pokémon series stands as one of Nintendo's oldest and most well-known franchises. With a unique blend of character designs and turn-based role-playing, the game introduced an entire generation to a distinctive gaming style, earning its place as the seventh most successful game of all time. Consequently, the franchise's current cultural significance comes as no surprise.

The community's embrace of "Fakémons," or self-made Pokémon, has reached a point where these creations feature prominently in entire fan-made games. This widespread enthusiasm paved the way for the exploration of generating new Pokémon through the application of deep learning. In recent years the generation of images has become more popular with the easy access to generative networks such as DALL-E, StyleGAN2 or GPT-3. Diffusion models describe one of the most popular architectures. This paper outlines the methodology employed for scraping and standardizing the data, along with the construction of the network architecture.

1 Introduction

- provide overview of the importance and possible applications of the 3D object detection (possibly extending the ones names by)
- describe the motivation for using machine learning in 3D object detection
- motivate why we want to reproduce the findings from
- outline the structure of the paper

2 Related Work

Generating Pokémon using deep learning is not a novel idea. Kleiber, 2020 used a Generative Adversarial Network (GAN) to create new Pokémon based on a dataset of 800 data samples. The presented results seen in Figure 1 can definitely be approved upon.

Figure 1: Results of Kleiber, 2020. The object somewhat resemble the colour scheme and form of a Pokémons. However, no image is recognizable as a complete design.

Using the same dataset Chambel, 2022 trained a Deep Convolutional Generative Adversarial Network (DCGAN) and achieved better results. The colour of the generated images is much closer to the often vibrant colour presented in the dataset. The results can be seen in Figure 2.

Figure 2: Results presented by Chambel, 2022.

3 Background

Ho *et al.*, 2020 present a new generative deep neural network architecture called the Denoising Diffusion Probabilistic Models (DDPM).

4 Data Acquisition and Preprocessing

4.1 Acquisition

There is no official Pokémon dataset that provides the extensive sample size typically required for training generative deep learning models (Yang *et al.*, 2023). Most of the available datasets offer only one image per Pokémon, resulting in approximately 1000 images in total. In comparison, the cifar10 dataset consists of 60,000 samples. Moreover, each Pokémon has a unique and easily distinguishable design, making it challenging to identify similarities between them. To address this limitation, I opted to create my own Pokémon dataset using web scraping.

I used web scraping to extract data from the Pokémon database, which lists all Pokémon along with various design variations for each. The Python packages BeautifulSoup and requests packages provide functions to parse the website into Hypertext Markup Language (HTML) and identify relevant tags. The national Pokédex which provides a list of all current Pokémon organized by Generation [1] represents the start point of the acquisition pipeline.

We collected links for each entry, simplifying the retrieval of all accessible images through a systematic naming convention. The link structure is as follows: "pokemondb.net/identifier/Pokémon." Changing the identifier granted access to two distinct datasets. The "artwork" identifier encompassed both official and alternative artwork, while the "sprites" section stored all in-game renditions of the Pokémon. In total, we obtained 26,896 unique images (22,825 sprites and 4,071 artworks). The term "unique" requires caution, as we intentionally included both male and female versions (even if differing by minor details) and normal and shiny versions (same Pokémon but with different color coding). Additionally, the database provides GIFs depicting the idle animation of the Pokémon. Each frame of these animations were added to the dataset. The back view of Pokémon were ignored since the goal was to generate images from the front.

The code for web scraping the images can be found here. Figure 3 shows an example of artworks and sprites for the Pokémon entry for Charizard.

Figure 3: Different renditions of the same Pokémon. Both samples from the artwork and sprite dataset are presented

Note that the trained model is a conditional diffusion model, thus the samples should be categorized in a logical scheme. There are multiple approaches on how the samples can be grouped. Typically, Pokémon are grouped based on their typing [2]. However, optimally similar images are grouped together. Pokémon belonging

[1] In the context of Pokémon, a "generation" refers to a specific group or series of Pokémon games released by Nintendo and Game Freak. Each generation introduces a new set of Pokémon species, game mechanics, and often a new region to explore.

[2] Typing refers to the elemental or thematic category (fire, water, ghost, fairy etc.) that a Pokémon belongs to. Each Pokémon has one or two types, which determine its strengths and weaknesses in battles. There are 18 different types in total, and each type has its own set of interactions with other types.

to the same type can differ drastically in regards to their appearance. Therefore, we opted to group samples based on the body from the Pokémon. This resulted in following classes.

Pokémon with:

1. only a head
2. a head and legs
3. fins
4. insectoid body
5. quadruped body
6. single pair of wings
7. multiple bodies
8. tentacles
9. base and legs
10. bipedal with tail
11. bipedal without tail
12. two or more pairs of wings
13. serpentine body
14. a head and arms

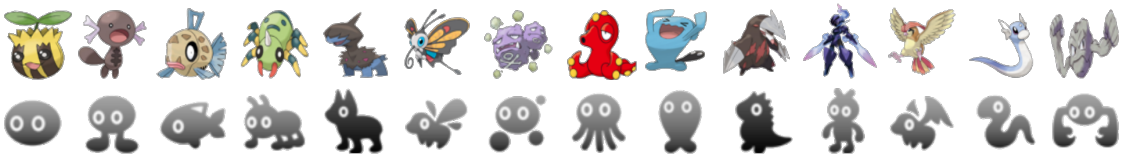

Figure 4: Different body types used to categorise the Pokémon.

4.2 Preprocessing

The next problem we encountered was the difference in their background, their image mode and type. At first we resize every image to 64x64. The original artwork and sprites were all transparent ".png", meaning they have no background and therefore a fourth channel encoding the transparency of the given pixel. This fourth channel resulted in partially transparent Pokémon and added unnecessary parameters to the network. Therefore we converted each of the "RGBA" images to uniform "RGB" images with a white background to keep their characteristic black outlining.
As the alternative artworks were collected from a variety of different artists there was no convention resulting in different file types (jpg, jpeg and vector png), backgrounds and image modes. To keep the format we established in the previous step we again removed the additional transparency channel and converted each image to png. Additionally, we detected each image which did not have a white background and adjusted it accordingly.
Moreover, was there a difference in coverage of the image. While the artworks used all available space of the image the sprites only used around 10% therefore exposing a lot of white background. This was apparent after training a few test epochs which resulted in solely white images with a small focus somewhere in the center (which was no surprise as the sprites represented the majority of the dataset). To counteract this, we located the focus point of the image (location of Pokémon) by removing each row/column only containing white pixels. Then to avoid morphing

the shape of the Pokémon we padded it to a square. At last we again resized the cropped image to get a uniform size of 64x64 pixels.
The code for making the images uniform can be found here.

An example for a sample before and after applying the uniform process can be seen in Figure 5.

Figure 5: A sample before and after applying the uniform process.

5 Model

The model employed was based on the DDPM presented by Ho *et al.*, 2020. Ho *et al.*, 2020 presents both an unconditional Model and a conditional one. For generating the images we only employed the conditional architecture. However, the architecture was slightly modified. The most significant change was the implementation of a scale factor which gives the ability to increase/decrease the amount of model parameters.

6 Model training

7 Results

The following describes the results reported on both the cifar10 dataset and our own Pokémon dataset. Since the quality of the generated image is somewhat subjective the Mean-square Error (MSE) during training is presented as well.

7.1 CIFAR-10

To validate the model pipeline we first trained the model on the cifar10 dataset since the model has already reported good results on this data. The CIFAR-10 dataset is a collection of 60,000 32x32 color images in 10 different classes, with 6,000 images per class. Each image belongs to one of the following classes:

1. airplane
2. car
3. bird
4. cat
5. deer
6. dog
7. frog
8. horse
9. ship
10. truck

Similarly, our model reported good results as well.

Figure (TBA) shows the decrease of MSE over time. Figure (TBA) shows the predicted images over time. As one can see the images predicted in the early stages of training were not reminiscent of the input data and mostly show either random noise or images containing only one colour. Similar to how the MSE decreased over time the quality of the images increased. The final images clearly depict objects of the respected class.

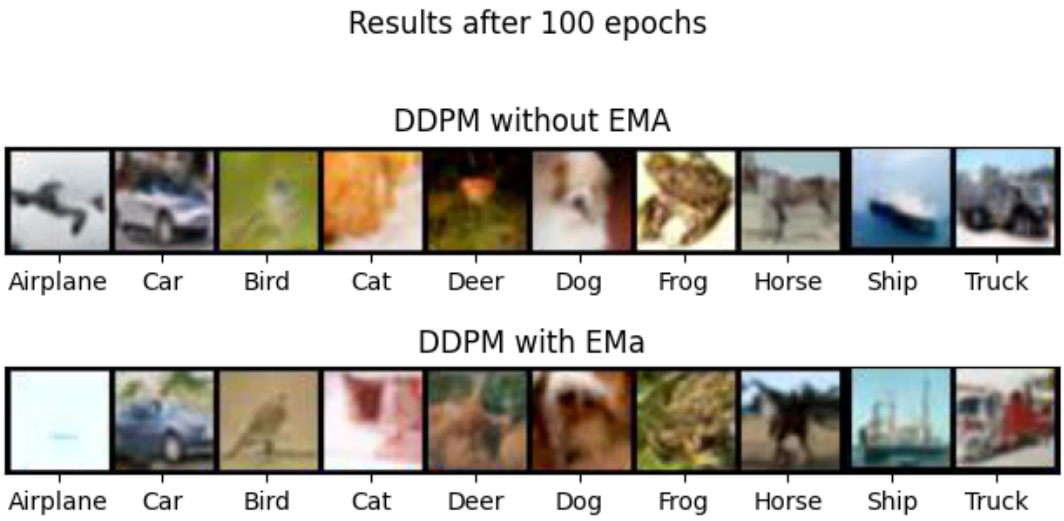

Figure 6: Generated images of DDPM

8 Discussion

Our model showed good results both on the benchmark dataset and our own created dataset. The diffusion model and the increased size of the dataset showed to positively influence the quality of the predicted images. Our images had a more recognizable design than previous studies (see Figure 1 & 2). This was expected since generative models are known to work best with larger datasets (Yang *et al.*, 2023). Additionally, DDPMs show to generate higher quality images than simple GANs (Guarnera *et al.*, 2023).

9 Conclusion

- How are the results of this study usable?
- Shortcomings of our study What was especially difficult?
- How can the results be improved upon?

- What might be further test that can be done to improve the generality?

References

Chambel, G. (2022). Generating realistic pokemons using a dcgan. https://medium.com/@goncalorrc/generating-realistic-pokemons-using-a-dcgan-331c7f75e211

Guarnera, L., Giudice, O., & Battiato, S. (2023). Level up the deepfake detection: A method to effectively discriminate images generated by gan architectures and diffusion models.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (pp. 6840–6851, Vol. 33). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

Kleiber, J. (2020). Pokegan: Generating fake pokemon with a generative adversarial network. https://medium.com/@jkleiber8/pokegan-generating-fake-pokemon-with-a-generative-adversarial-network-f540db81548d

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., & Yang, M.-H. (2023). Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, *56*(4), 1–39.