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 Wang *et al.*, 2023)
- describe the motivation for using machine learning in 3D object detection
- motivate why we want to reproduce the findings from Wang *et al.*, 2023
- outline the structure of the paper

2 Related Work

- explain findings by Wang *et al.*, 2023 further and the other papers named in the study and the current state of the art models

3 Background

- explain the fundamentals of 3D object detection and the use of machine learning to solve the most common problems

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 (TODO: find source for generative models needing a large dataset). Most of the available datasets offer only one image per Pokémon, resulting in approximately 1000 images in total. In comparison, the cifar10 dataset (TODO: add href to cifar10) consists of 60,000 samples. Complicating matters, each Pokémon boasts 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 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. Event versions, such as differently

[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.

clothed "pikachu," were retained, while sprites depicting Pokémon from behind were excluded.

The code for web scraping the images can be found here. Figure 1 shows an example of artworks and sprites for the Pokémon entry for Charizard.

Figure 1: Different renditions of the same Pokémon. Both samples from the artwork and sprite dataset are presented

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 2.

Figure 2: A sample before and after applying the uniform process.

5 Model

- describe the models architecture and our possible changes to it
- briefly describe the other models as well and in how they differ from our model

6 Model training

- describe the training loop for the model

7 Results

- describe the results
- showcase of the performance of each models comparing various metrics
- also showing the training comparison

8 Discussion

- discuss the findings of the results:
- Where the results presented by Wang *et al.*, 2023 replicable?
- In how far did our results differ from the the ones presented Wang *et al.*, 2023?
- What might be the reason for the differences?

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

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. (2016). 3d semantic parsing of large-scale indoor spaces. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). Nuscenes: A multimodal dataset for autonomous driving. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., & Nießner, M. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes.

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, *32*(11), 1231–1237.

Song, S., Lichtenberg, S. P., & Xiao, J. (2015). Sun rgb-d: A rgb-d scene understanding benchmark suite. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 567–576. https://doi.org/10.1109/CVPR.2015.7298655

Wang, Z., Li, Y., Chen, X., Zhao, H., & Wang, S. (2023). Uni3detr: Unified 3d detection transformer.