OPEN PROGRAM - TEXT TO IMAGE

Maxwell Ernst

Contents

Introduction

This report aims to delve into the field of text-to-image generation and explore its potential applications. The project involves an exploration of Generative Adversarial Networks (GANs), data augmentation techniques, and the development of a Python application to generate images using diffusion models. By undertaking this project, I aimed to enhance my understanding of text-to-image generation and contribute to the advancements in this area. This report will present the methods employed, the results obtained, and offer insights into the future possibilities and implications of text-to-image technology.

In recent years, text-to-image generation has gained significant attention due to its potential applications in various domains such as art, design, advertising, and entertainment. Being able to generate high-quality images from textual descriptions opens up new avenues for creativity and automation. With the advancements in machine learning and deep learning techniques, it has become possible to train models to understand and translate text into visually appealing and coherent images.

Goal

The objective of this project is to explore and expand my knowledge of text-to-image generation. Although I have experimented with text to image AI services like Midjourney, I aim to delve deeper into the field and gain a deeper understanding of the intricacies involved in generating images from textual descriptions.

Research

Text-to-image synthesis is the task of generating images from natural language descriptions. It is a challenging and exciting problem that has many potential applications, such as visual storytelling, content creation, and image editing. In this research, we will review some of the recent advances in text-to-image models, focusing on two novel approaches: GANs and Diffusers.

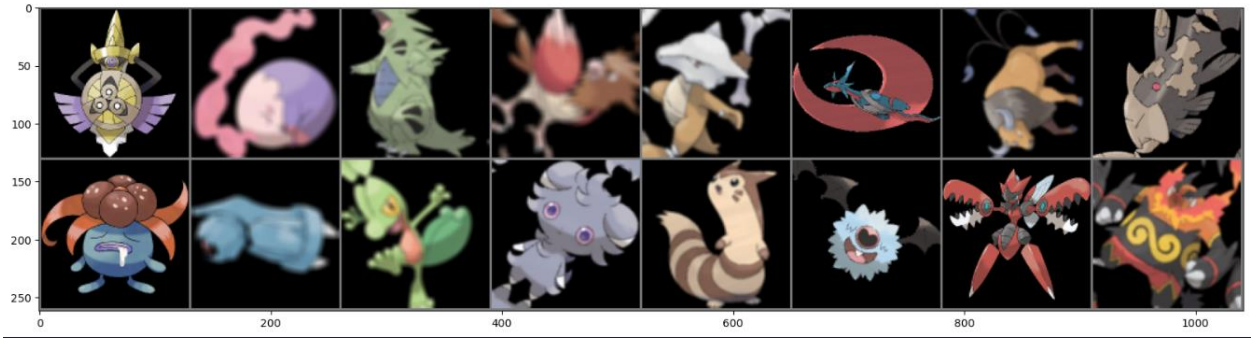
GANs, or generative adversarial networks, are a type of neural network that consists of two components: a generator and a discriminator. The generator tries to produce realistic images from text, while the discriminator tries to distinguish between real and fake images. The generator and the discriminator compete with each other, improving their performance over time. GANs have been widely used for text-to-image synthesis, achieving impressive results on various datasets. Some examples of GAN-based text-to-image models are DALL-E 2[1], VQ-GAN+CLIP[2], and AttnGAN[3].

Diffusers, or diffusion models, are another type of neural network that generates images from text by reversing a diffusion process. The diffusion process gradually transforms an image into random noise by adding Gaussian noise at each step. The diffusion model learns to reverse this process, starting from noise and removing noise at each step, conditioned on the text input. Diffusers have been shown to produce high-fidelity and diverse images from text, without suffering from mode collapse or blurry artifacts. Some examples of Diffuser-based text-to-image models are Imagen[4], Latent Diffusion Models[5], and Text-to-Image Diffusion Models[6][6].

Text-to-image synthesis is an active area of research that has made significant progress in recent years. However, there are still many challenges and open questions to be addressed, such as improving the image-text alignment, handling complex and abstract concepts, ensuring ethical and responsible use of the technology, and evaluating the quality and diversity of the generated images. We hope that this research will provide a useful overview of the current state-of-the-art methods and inspire future work in this domain.

Data

The data I worked with was mainly during the first tutorial I did on the Pokemon data set found on Kaggle. To begin, I employed data augmentation techniques using the Albumentations library. These techniques involved resizing, cropping, and transforming the Pokemon images. By doing so, I created augmented versions of the images that captured various perspectives and variations, making the dataset more diverse and rich.



After completing the data augmentation I prepared the images and split the images into train and test sets.

Modeling

I not only explored traditional GAN modeling techniques but also experimented with pre-trained models available on Hugging Face. This allowed me to leverage the power of pre-existing models and assess their performance in the context of text-to-image generation.

For generating the Pokemon images, I implemented a GAN model comprising the Generator and Discriminator networks. The Generator utilized a random noise vector as input and generated synthetic images that closely resembled real Pokemon. The Discriminator, on the other hand, was responsible for distinguishing between real and fake images.

To train the GAN model, I employed the Adam optimizer and the binary cross-entropy loss function. By monitoring the losses of both the Generator and Discriminator, I was able to evaluate the convergence and stability of the model, ensuring that it was gradually improving in its ability to generate realistic Pokemon images.

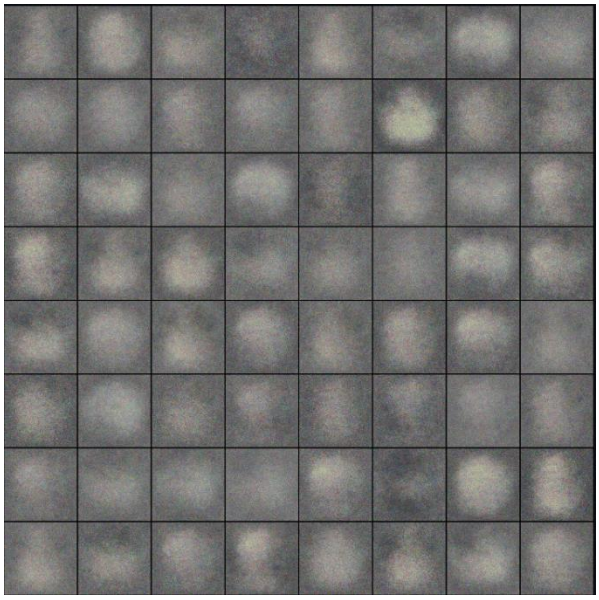
Throughout the training process, I regularly examined the generated images to observe the progress of the Generator. This visual inspection provided insights into how the model was evolving and generating increasingly realistic Pokemon images. Furthermore, I conducted experiments with different modeling techniques and adjusted hyperparameters to enhance the quality of the generated images.

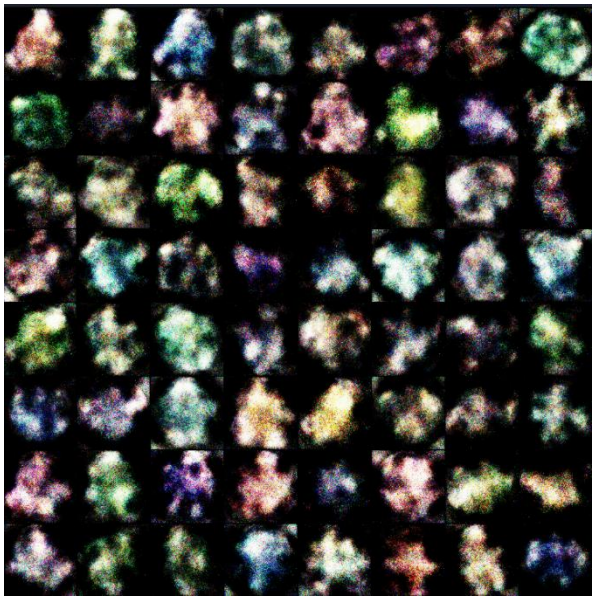
In conclusion, the "Model" section of this report showcases the successful utilization of data augmentation techniques and GAN modeling in generating Pokemon images. Moreover, I extended my exploration by incorporating pre-trained models from Hugging Face, broadening the scope of the project and evaluating the performance of these models. However, it's worth noting that due to the limitations of training time on my CPU, the results could have been further improved by running the notebook on a GPU. Nonetheless, the outcomes obtained are indicative of the potential of text-to-image generation in producing compelling and visually appealing results.

Results

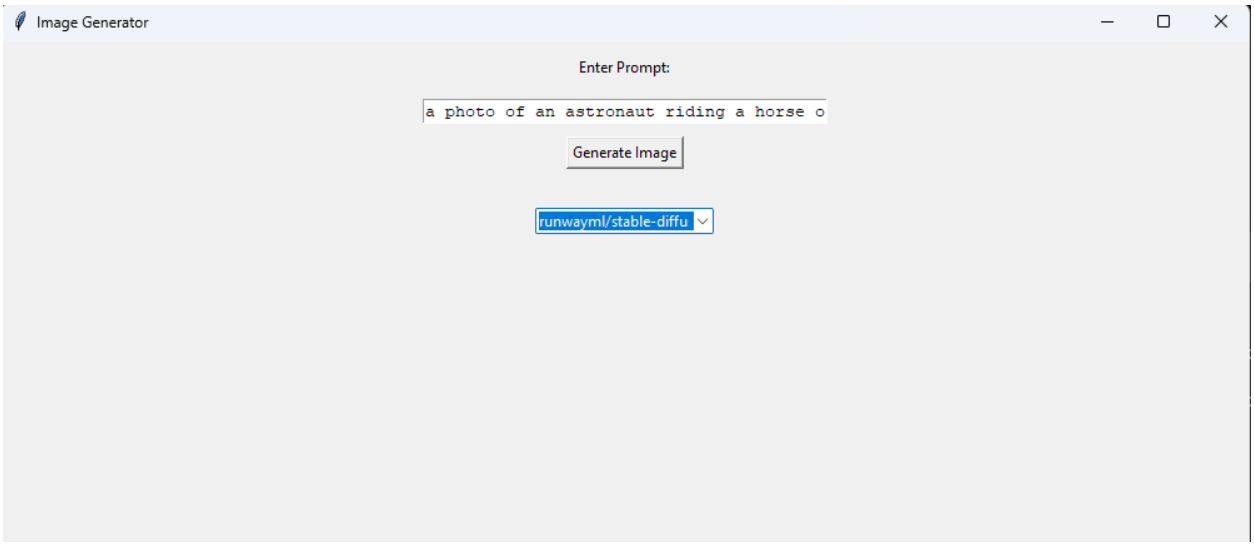
The results of the project are outlined below:

1. Completion of GANs Tutorial: I successfully completed a tutorial on Generative Adversarial Networks (GANs) using the Pokemon dataset. The tutorial covered important aspects such as data augmentation and modeling techniques. Additionally, I performed fine-tuning on the models to achieve the best possible results in terms of image generation.





2. Development of a Python Application: I developed a Python application utilizing the Tkinter library, and streamlit. This application enables users to generate images using a selection of diffusion models available on Huggingface. Users have the ability to choose from multiple models and compare the generated results. This interactive interface provides a convenient way to explore and assess the capabilities of different models for image generation. I used Tkinter to create a local application and streamlit to deploy as a browser application. There is also a version for Mac.



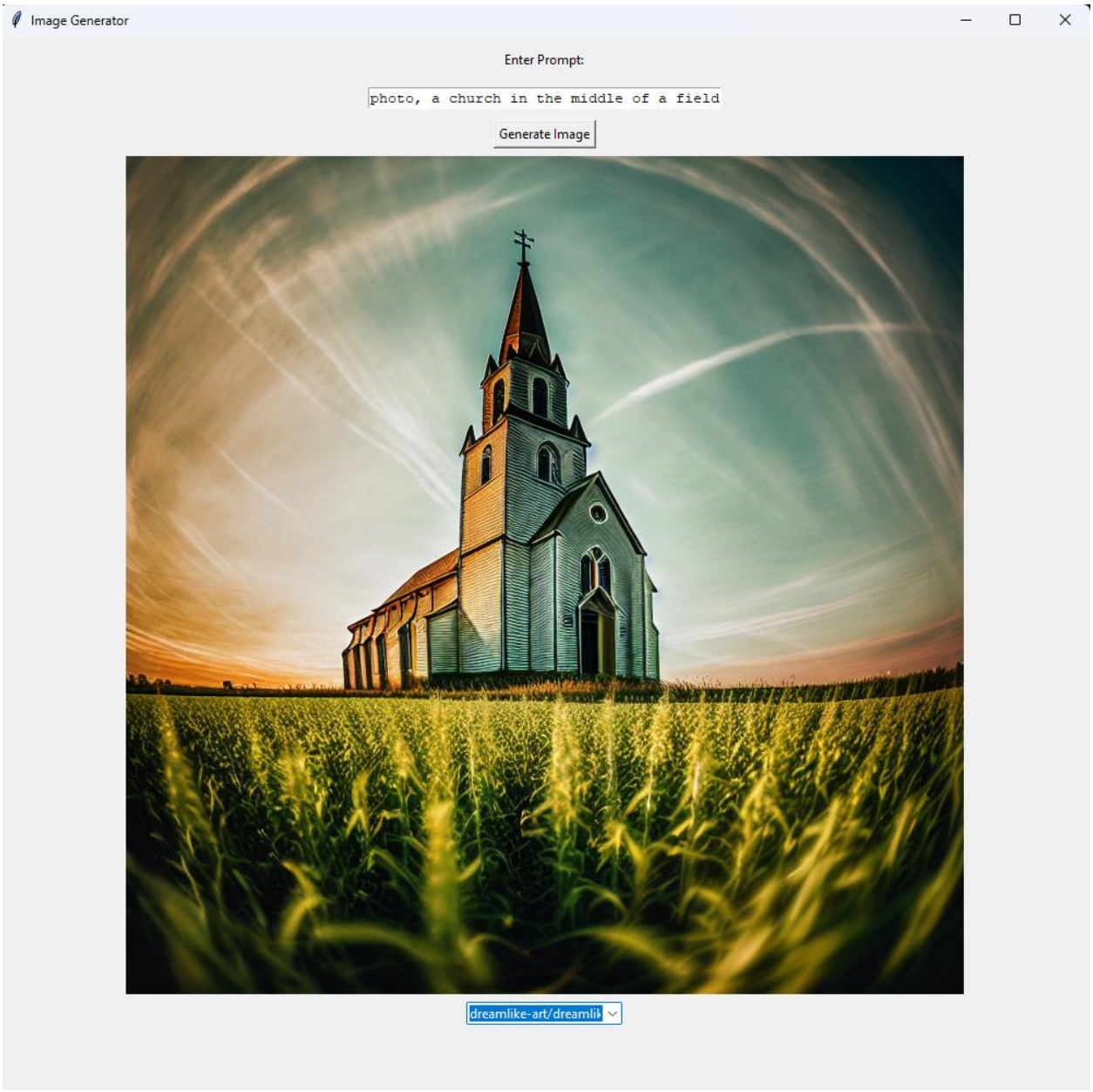
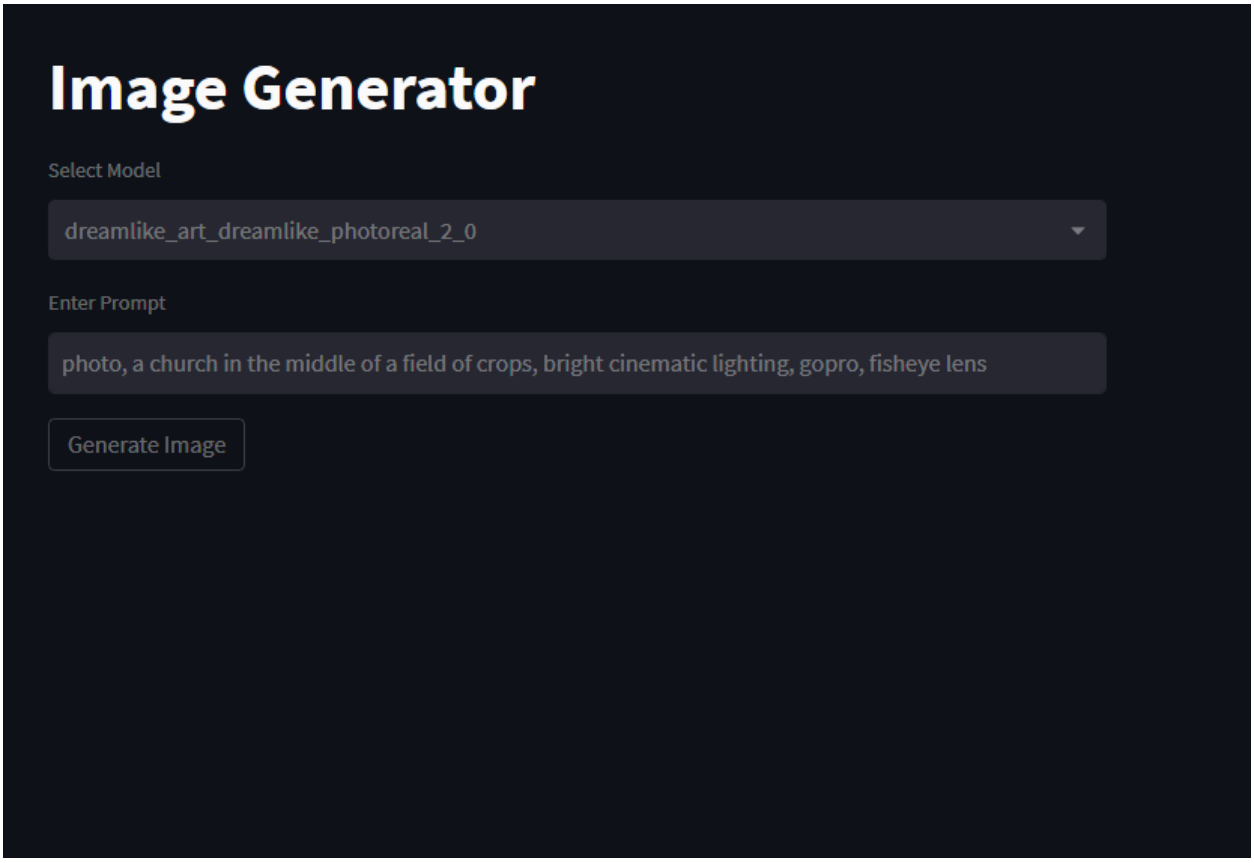


Image Generator
Enter Prompt:
photo, a church in the middle of a field
Generate Image
dreamlike-art/dreamlik



Overall, the completion of the GANs tutorial and the development of the Python application have allowed for a comprehensive exploration of text-to-image generation techniques and facilitated the comparison of results obtained from various models.

Conclusion

The primary objectives of this project were to explore GANs, implement data augmentation techniques, and develop a Python application that allows users to generate images using diffusion models. By completing a tutorial on GANs with the Pokemon dataset, I acquired hands-on experience in applying these techniques to generate realistic images. Additionally, fine-tuning the modeling process allowed me to optimize the results and improve the quality of the generated images.

Furthermore, the development of the Python application using the Tkinter library provided an interactive platform for users to experiment with different diffusion models. This not only facilitated the comparison of results but also enabled users to gain a better understanding of the capabilities and limitations of various models in the context of text-to-image generation.

In conclusion, this report presents the methods, techniques, and results obtained during the exploration of text-to-image generation. By completing a GANs tutorial, implementing data augmentation techniques, and developing a Python application, I have gained valuable insights into the process of generating images from textual descriptions.

References

[1]: Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, Ilya Sutskever. DALL·E: Creating Images from Text. arXiv preprint arXiv:2102.12092 (2021)

[2]: Patrick Esser, Robin Rombach, Björn Ommer. Taming Transformers for High-Resolution Image Synthesis. arXiv preprint arXiv:2012.09841 (2020)

[3]: Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, Xiaodong He. AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1316-1324 (2018)

[4]: Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. arXiv preprint arXiv:2205.11487 (2021)[5]: Jonathan Ho et al. Latent Diffusion Models for Text-to-Image Generation. arXiv preprint arXiv:2106.05048 (2021). [6][6]: Robin Zbinden. Implementing and Experimenting with Diffusion Models for Text-to-Image Generation. Master's Thesis (2022).