Text-Based image editing using diffusion models
Machine Learning 2022 - Project Report
CERVERA Matthieu

Abstract

This project aims to create an edited image given a base image and a target text as input (e.g. an image of a cat and a text like 'a cat wearing a hat'). It builds on a pre-trained text-to-image diffusion model and applies text semantic edits to the image, such as composition or posture changes, etc. In this project I tried to implement and understand the method of the paper Imagic: Text-Based Real Image Editing with Diffusion Models B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, M. Irani [3]. All my work is based on this particular paper, even if other papers will be cited throughout the report.

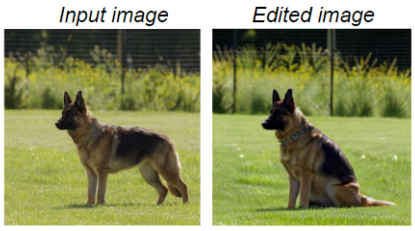


Figure 1. My implementation's edited image

.

Introduction

Applying non-trivial semantic edits to real photos has long been an interesting task in image processing. It has attracted considerable interest in recent years, enabled by the considerable advancements of deep learning based systems. Image editing becomes especially impressive when the desired edit is described by a simple natural language text prompt, since this aligns well with human communication. In this report, I propose an implementation of a semantic image editing method. Given only an input image to be edited and a single text prompt describing the target edit, the method can perform sophisticated non-rigid edits on real high-resolution images. The resulting image outputs align well with the target text, while preserving the overall background, structure, and composition of the original image.

In this report, I'll summarize the method and show that even this complex method can be implemented using less ressources and a less performant neural network while still allowing complex non-rigid edits on a single real input image, and still preserving the image's overall structure and composition.

Problem Definition

Diffusion models are a family of generative models that has recently gained traction, as they advanced the state-of-the-art in image generation. The goal here is to use this type of models to generate an image edited according to a text prompt. The core premise of these models is to initialize with a randomly sampled noise image $x_T \sim \mathcal{N}(0, I)$, then iteratively refine it in a controlled fashion, until it is synthesized into a photorealistic image x_0. Each intermediate sample x_t (for $t \in \{0, ..., T\}$) satisfies

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon_t$$

with $0 = \alpha_T < \alpha_{T-1} < ... < \alpha_1 < \alpha_0 = 1$ being hyperparameters of the diffusion schedule, and $\epsilon_t \sim \mathcal{N}(0, I)$. Each refinement step consists of an application of a neural network $f_\theta(x_t, t)$ on the current sample x_t, followed by a random Gaussian noise perturbation, obtaining x_{t-1}. The network is trained for a simple denoising objective, aiming for $f_\theta(x_t, t) \approx \epsilon_t$. This leads to a learned image distribution with high fidelity to the target distribution, enabling stellar generative performance.

By incorporating knowledge from large language models or hybrid vision-language models, some text-to-image diffusion models have unlocked a new capability – users can generate realistic high-resolution images using only a text prompt describing the desired scene.

Related Work

Recent advancements in image synthesis have led to the use of pretrained generative adversarial networks (GANs)

for a variety of image manipulation techniques, including optimization-based, encoder-based, and model-per-input methods. Other deep learning-based systems, such as diffusion models, have also been used for image editing tasks with promising results. SDEdit [5] and DDIB [10] are examples of these diffusion models, which use intermediate noise and DDIM inversion to edit images. Textual Inversion and Dream-Booth [8] synthesize novel views of a given subject given 3–5 images of the subject and a target text (rather than edit a single image), with DreamBooth requiring additional generated images for fine-tuning the models. Imagic method provide the first text-based semantic image editing tool that operates on a single real image, maintains high fidelity to it, and applies non-rigid edits given a single free-form natural language text prompt."

Methodology

1. Imagic Method

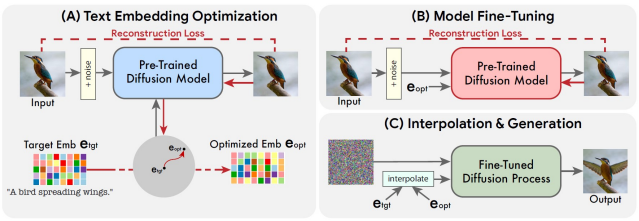


Figure 2. **Schematic description of Imagic**.

The paper I rely on adapts the text-to-image diffusion models in their work to edit real images instead of synthesizing new ones. They do so in a simple 3-step process, as depicted in Figure 2: Given a real image and a target text prompt: (A) They first optimize a text embedding so that it results in images similar to the input image. (B) Then, they fine-tune the pre-trained generative diffusion model (conditioned on the optimized embedding) to better reconstruct the input image. (C) Finally, they linearly interpolate between the target text embedding and the optimized one, resulting in a representation that combines both the input image and the target text. This representation is then passed to the generative diffusion process with the fine-tuned model, which outputs the final edited image.

2. Imagic Implementation

In the paper they implement the method using the generative diffusion model Imagen [9]. Using classifier-free guidance [2], Imagen constitutes a powerful text-guided image generation scheme.

They optimize the text embedding using the diffusion model and the Adam [4] optimizer for 100 steps and a fixed learning rate of 1e-3. They then fine-tune the diffusion model by continuing Imagen's training for 1500 steps with the input image, conditioned on the optimized embedding. In parallel, they also fine-tune another part of the diffusion model using the target text embedding and the original image for 1500 steps, in order to capture high-frequency details from the original image. This entire optimization process takes around 10 minutes per image on two massive computational TPUv4 chips.

3. My Implementation

I based my implementation on Justin Pinkney's one published on github [6]. Obviously it's not the same as the one they used in the paper, because of computational power issues. They used the diffusion model Imagen which is not publicly available and they fine tuned on 2 TPU v4 chips which are resources I can't afford.

So I used the open source stable diffusion v1-4 model [7] from hugging face. I started with a small GPU and was not able to fine-tune the generative model. I finally changed to a NVIDIA A100 40Gb so that I could compute the whole pipeline. I optimized the text embedding for 1500 steps and I then fine-tuned the diffusion model for 1500 steps. Sometimes I got qualitatively better results with 2000 steps for each optimization but it wasn't really a consistent result.

Finally I implemented two quantitative scores to rate my results.

Evaluation

To experiment my implementation, I mostly used images from Textual editing benchmark (TedBench). These allowed me to compare qualitatively my results to other methods presented above. I first experimented the model with images from celebrities, trying to make some edits on their faces. Then I tested with TedBench benchmark images. After I compared my model to others, I was interested in implementing a pertinent quantitative evaluation of my results, as they did in the main paper. I also did encounter some failure cases, and I tried to find some explanations.

1. Qualitative evaluation

a. First results and memory issue

My first steps were to re-use the same image and input text as in the code I found from Justin Pinkney. The desired edit was to add a smile on Barack Obama's official picture. Unfortunately I had GPU memory issues. Indeed, the virtual machine I used had a too small GPU memory for my implementation to work. Therefore I could not fine-tune the model. I still managed to optimize the embedding to resemble the input image, so my best result shows exciting edited image. Fig.3 shows a generated image that hasn't fully reproduced the input image, buts still has a good angle.

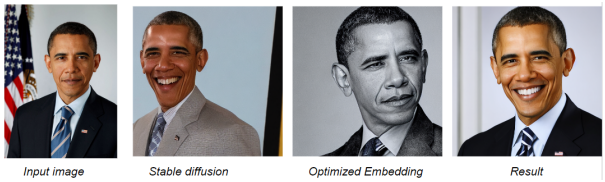


Figure 3. Target text : "Obama smiling with a big grin"

b. Fine-tuning the model

With the new GPU, I was able to fine-tune my model. I optimize the model parameters while freezing the optimized embedding. The generative diffusion model is trained to fully recreate the input image at the optimized embedding. As shown in Fig.4 , the images generated by the fine-tuned model at the optimized embedding are very similar to the input images. There are some colors, shapes and textures that remain imperfect dependently on the input image, but it's already a good start to try to interpolate the embeddings in order to get the desired edits.

Figure 4. Comparison between input Image (top) and Image generated by the fined-tuned model at the optimized embedding (bottom)

c. Embedding interpolation and results

When the whole model was fine-tuned, I needed to advance from the optimization embedding in the direction of the target text embedding, using the interpolation equation for $\eta \in [0, 1]$.

$$e = \eta e_{text} + (1 - \eta) e_{optimized}$$

e_{text} being the base embedding of the target text, and $e_{optimized}$ the optimized embedding to resemble more the input image. When interpolate the embedding using the same seed for the image generation, the result is bluffing (Fig.5). We can see that, more η is close to 1, more the edit is applied (the car becomes more and more yellow) and more η is close to 0, more the original input is preserved in the generated image (the orignal image background is transformed by the edit). This is a really exciting result, because it means that by interpolating two objects from the embedding space, the result is coherent to my expectations. This space of embeddings (a space with high dimensional tensors) seems to be close to human expectation, and we could even think of a parallelism between the embedding space and the human thoughts : when you say the word 'apple', your mind has its own representation of this word, representation that can change into an image, or into close words, synonyms, exactly as a NLP or diffusion model works.

Figure 5. Embedding interpolation - target text : "a yellow car"

I wanted to try the whole pipeline with an image of another person than Barack Obama. Unfortunately, I figured out that the traits of an unknown person are difficult to reproduce correctly. Furthermore, I was the qualitative evaluator of my results and the slightest defect on a human generated face is really disturbing. I got the idea to use well-known people. That's why I tried to edit a photo of George Clooney, to add him sunglasses or a hat : Fig.6 shows the whole pipeline results for target text 'George Clooney wearing sunglasses'.

Figure 6. Whole process using a single seed - target text : 'George Clooney with sunglasses'.

Other results are showed in Fig.7, where I tested several edits with 2 different seeds each time. We can see obvious differences with the input image, however the edit stays close to the expected result and close to the input image also.

Figure 7. Edit of George Clooney with multiple target texts and two seeds.

2. Comparison with other methods

I compared the results of my method to the current leading techniques that use a single image input, and that apply an edit based on a target text. To do so, I used the results comparing Imagic and other methods published in the paper. I added my method results next to Text2LIVE, DDIB, SDEdit and Imagic's one provided by the paper. We can see that Imagic results are way better than any of other methods (including mine). However, compared to other methods my images can preserve a bit the input image while applying the edit. Fig.8 is completed by Fig.10 in the appendix.

Figure 8. Comparison of my implementation with other methods

3. Quantitative evaluation

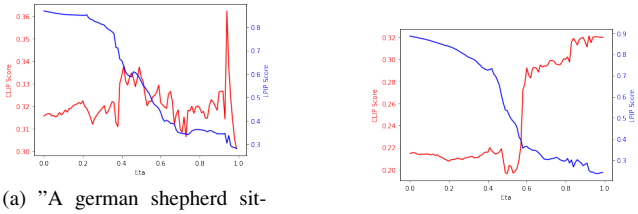
Interesting and worthwhile quantitative results are hard to find in this kind of project. Indeed, the most important results are the qualitative ones. But pertinent quantitative results could always serve this kind of project. I got interested in two different scoring methods that are used in the paper : CLIP [1] and LPIPS [11]. Both methods rely on neural networks.

a. CLIP Score

CLIP Score is a scoring method that uses a ResNet image encoder and a text encoder. It rates the resemblance of the input image with a target text. One can use it in order to measure the resemblance of the generated output with the target text. It is the value that enables us to quantify if the edit is done as desired. I used *Hugging Face* implementation, as for the stable diffusion process. It seems that the CLIP score varies usually between 0.2 and 0.35 and the higher it is, the more faithful is the edited image to the target text.

b. LPIPS Score

Learned Perceptual Image Patch Similarity (LPIPS) metric allows us to measure the similarity between two images by evaluating the distance between image patches. This metric is useful in our case to check if the generated image stays close to the original input image. If the edit does affect the input image, LPIPS score will be close to 1 and on the other hand, if LPIPS score is close to 0 the generated image has a higher fidelity to the input image, .



(a) "A german shepherd sitting"

(b) "A zebra"

Figure 9. CLIP and (1-LPIPS) functions of η

I computed $(1 - LPIPS)$ and CLIP score for $\eta \in [0, 1]$. I used the picture of the dog with the edit "a german shepherd sitting" and the picture of the horse with the edit "A zebra". Fig.9 shows that with embedding interpolation, we can control the edit and the input image fidelity. With several tests, a interval of η emerges where generated image has a good alignment with the target text and still a high fidelity to the input image. Unlike the paper, I found that interval was $\eta \in [0.4, 0.7]$, but the results of the paper were really better and the interval they found ($\eta \in [0.6, 0.8]$) is more accurate than mine.

However, while these scores provide a good idea of text or image alignment on average, they are not really accurate measures (see the CLIP score fluctuation in Fig.9. I figured that the values can really differ for different - but still close qualitatively - inputs. Therefore the performance of the edit can't be fully rated quantitatively for now, and a qualitative evaluation as the user study done in the paper is much more reliable for now.

4. Limitations

a. Failure cases

Obviously, I had several failure cases that came back regularly while experimenting some edits. Fig.11 shows most of the classic failure cases that I faced. Sometimes, the edit is not applied to the input image with a too small η but when we increase η, there lacks consistency with the input image. I also got a dog turning to a sheep with the target-text "A german shepherd sitting". An ambiguous target-text for the diffusion model mostly fails the edit. I think that the base image and the complexity of the desired edit do change the quality of the result.

b. Stable diffusion limitation

We can perhaps try to explain these failures also because *Imagen* seems to be way more efficient than stable diffusion. *Imagen* generated images have an enormous consistency with difficult ambiguous and long target-texts whereas *Stable diffusion* could generate some failed images with text as simple as 'a zebra'.

Conclusion

The paper's technique I tried to implement is the first text-based semantic image editing technique that allows for complex non-rigid edits on a single real input image, while preserving its overall structure and composition. I was able to obtain quite satisfactory results if we consider the difference in computing capacity. However, the results of the paper are way better, and are among the best research results currently available. This paper also demonstrate a semantically meaningful linear interpolation between two text embedding sequences, uncovering strong compositional capabilities of text-to-image diffusion models.

For future ideas I thought of automatise the process of finding the best seed then the best η in order to generate a more accurate edited image without having to manually pick. However, the scores I implemented are not suited to reliably choose η. So a first step might be to improve the quantitative scoring methods. Lots of art-related models (for music, or images) are really missing precise quantitative scoring as it is a really difficult task.

References

[1] Jack Hessel et al. *CLIPScore: A Reference-free Evaluation Metric for Image Captioning*. 2021. arXiv: 2104.08718 [cs.CV].

[2] Jonathan Ho and Tim Salimans. *Classifier-Free Diffusion Guidance*. 2022. arXiv: 2207.12598 [cs.LG].

[3] Bahjat Kawar et al. *Imagic: Text-Based Real Image Editing with Diffusion Models*. 2022. arXiv: 2210.09276 [cs.CV].

[4] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014).

[5] Chenlin Meng et al. *SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations*. 2021. arXiv: 2108.01073 [cs.CV].

[6] Justin Pinkney. *Imagic using Stable Diffusion*. 2022. URL: https://github.com/justinpinkney/stable-diffusion/blob/bcf01b15796540246896ff58eef75f109a220992/notebooks/imagic.ipynb.

[7] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. DOI: 10.48550/ARXIV.2112.10752. URL: https://arxiv.org/abs/2112.10752.

[8] Nataniel Ruiz et al. *DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation*. 2022. arXiv: 2208.12242 [cs.CV].

[9] Chitwan Saharia et al. *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*. 2022. arXiv: 2205.11487 [cs.CV].

[10] Xuan Su et al. *Dual Diffusion Implicit Bridges for Image-to-Image Translation*. 2022. arXiv: 2203.08382 [cs.CV].

[11] Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. 2018. arXiv: 1801.03924 [cs.CV].

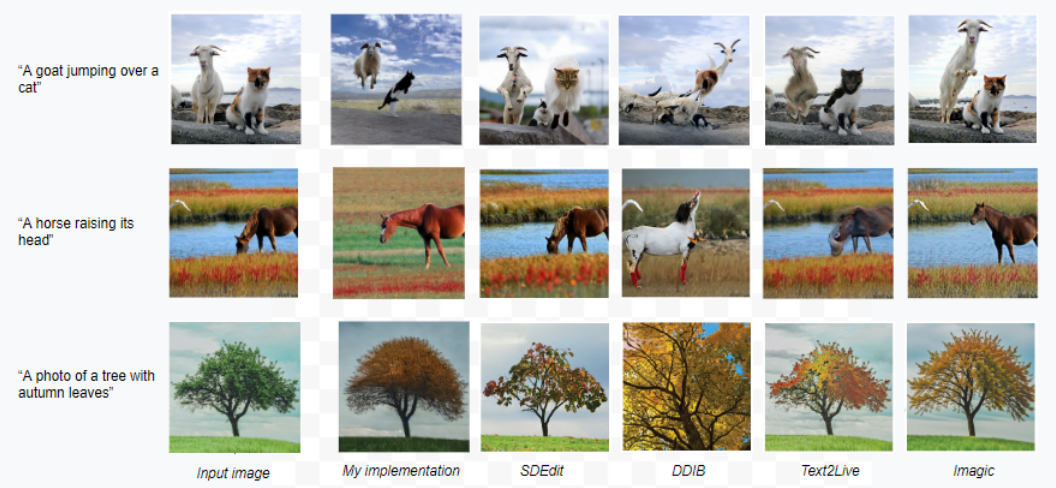


Figure 10. Comparison of my implementation results with other methods

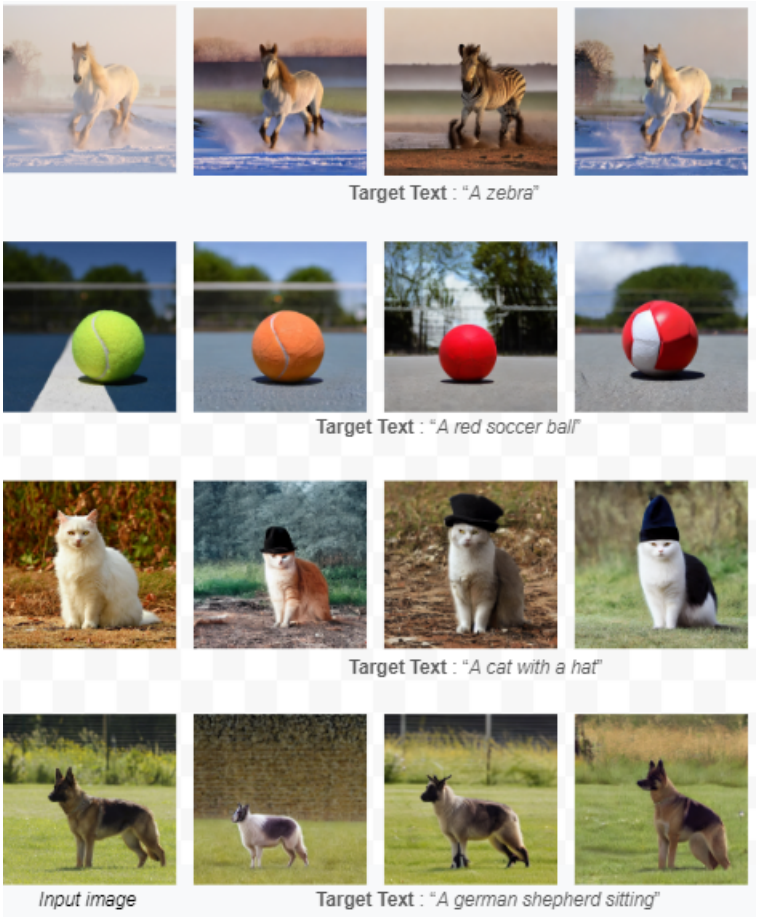


Figure 11. Failure cases