MASTER AI IMAGE GENERATION USING STABLE DIFFUSION

EXPERT ACADEMY

COURSE CONTENT

- Part 1: Basics about Stable Diffusion
- Part 2: Prompt Engineering
- Part 3: Fine-tuning
- Part 4: Image-to-image
- Part 5: Inpainting
- Part 6: ControlNet

- Only the basic intuition about how it works
- Pre-requisites
  - Programming logic (not mandatory)
  - Basic Python programming (not mandatory)

WHAT IS STABLE DIFFUSION?

- **Stable Diffusion** is a deep learning model for image generation, released in 2022.
- It belongs to a class of models called *diffusion models*.
- They are generative models, which means they are designed to create new data similar to the data it was trained on. In the case of Stable Diffusion, the data are images.
- It is based on a particular type of diffusion model called **Latent Diffusion**, proposed in the paper *High-Resolution Image Synthesis with Latent Diffusion Models*.
- Diffusion models are machine learning systems trained to reduce random Gaussian noise.

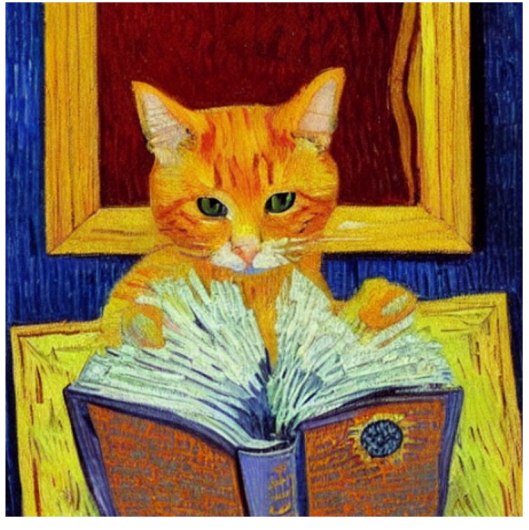
Prompt: *"a photograph of an astronaut riding a horse"*

WHAT IS STABLE DIFFUSION?

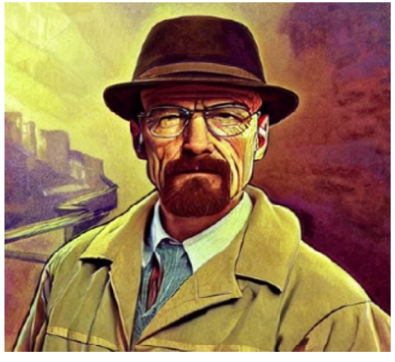
- These models generates high quality images that did not exist before.
- Images can be produced from keyword correlations. The model manages to correct the artistic style, faces and shadows, and blends them together in an aesthetically pleasing way.
- This technique gained popularity due to its ability of generationg good images using creative concepts and unusual combination of keywords.

Prompt: *“an orange cat reading a book, painted by van gogh”*

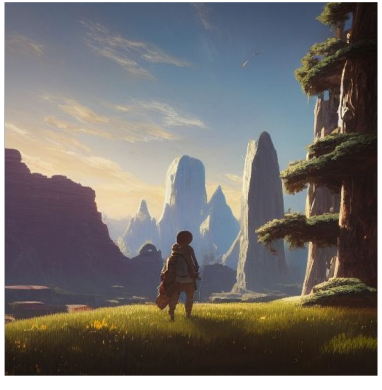
Model: v1.5



SOME EXAMPLES OF WHAT STABLE DIFFUSION CAN DO

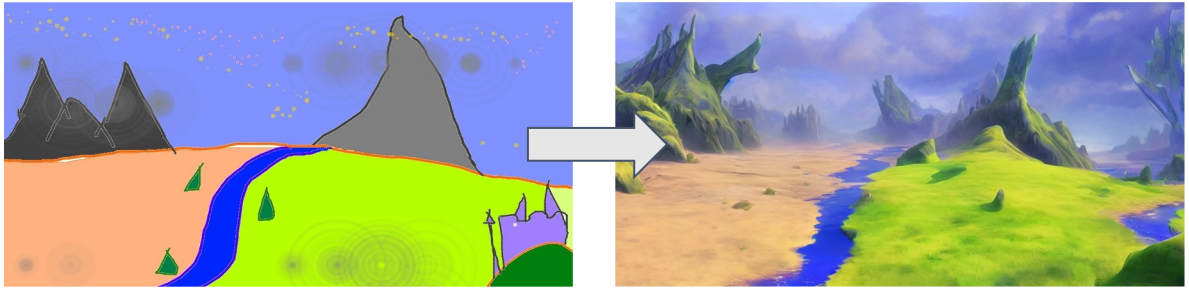




STABLE DIFFUSION - FEATURES

Text-guided image-to-image
It allows using an initial image as input to condition the generation.



Inpainting
It allows selecting a specific part of the image to change the class/concept, or even removing it from the scene.

Image source: huggingface.co/docs/

STABLE DIFFUSION - FEATURES

Fine-tuning - customized training used to “insert” a new concept (person, object, style, etc.) to the model and thus obtain stylized images from the provided prompt

photo of *zwx* person, in an armor, realistic, visible face, colored, detailed face, ultra detailed, natural lighting

painting of *zwx* person in star wars, realistic, 4k ultra hd, blue and red tones

photo of *zwx* person as an astronaut, natural lighting, frontal face, closeup, starry sky in the background

photo of *zwx* person in a western movie

STABLE DIFFUSION - FEATURES

Super-Resolution
Diffusion model for upscaling images, i.e. increase its resolution and make it sharper.

Outpainting / Image extension
Allows you to extend image regions that have been cropped or do not exist.

TYPES OF IMAGE GENERATION

Unconditional image generation: the model generates images without any additional condition like text or image. You will get images similar to those provided in the training set.

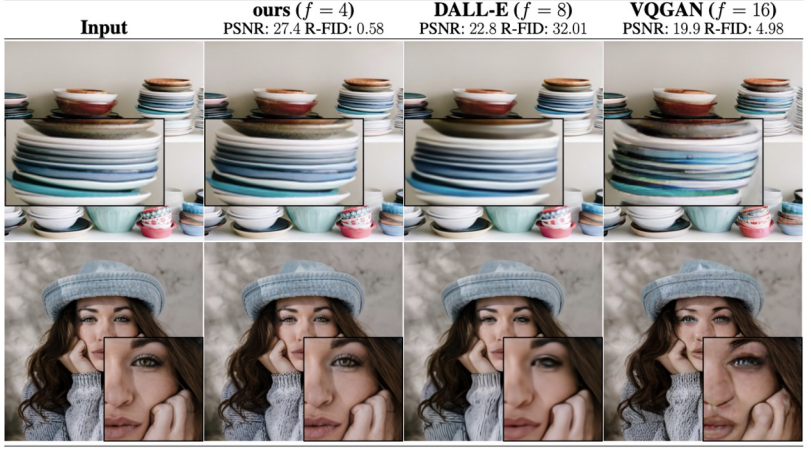
- The generation of images conditioned by text is known as ***text-to-image, or text2img***.
    - The **prompt** is converted into embeddings that are used to condition the model to generate an image from noise.
- The generation of images based on other image is known as ***image-to-image, or img2img.***
    - In addition to the text prompt, it allows sending an initial image to condition the generation of new images. You have more control over the final composition.

STABLE DIFFUSION - ADVANTAGES

- It is open source. Many enthusiasts develop free and powerful tools.
- You can run it on simple machine configurations. Although GPU usage provides more speed, it doesn't need to have much RAM to run it because of its efficient architecture. Therefore, it democratizes the use of modern solutions for generating images from text.

STABLE DIFFUSION - ADVANTAGES

Another key advantage of Stable Diffusion compared to previous methodologies is the ability to **scale** images with quality.



Source: official paper https://arxiv.org/abs/2112.10752

It allows:

- working at a level of compression that provides more **faithful and detailed** reconstructions than previous models;
- **working efficiently for high-resolution generation of large images.**

STABLE DIFFUSION – COMPARISON AND ADVANTAGES

- Diffusion models like Imagen (from Google) and DALL-E (from Open AI) work in pixel space. They use some tricks to make the model faster.
- Stable Diffusion was developed to solve the speed problem. It is a latent diffusion model.
- Instead of operating in high-dimensional image space, it first compresses the image into latent space.
- The latent space is 48 times smaller, so it is much faster.
- Compared to DALL-E 2 and Imagen, the Stable Diffusion model is much smaller. While DALL-E 2 has around 3.5 billion parameters and Imagen has 4.6 billion, the first Stable Diffusion model only has 890 million parameters.

STABLE DIFFUSION – CONTEXT

- The Stable Diffusion algorithm was developed by Compvis (the Computer Vision research group at the Ludwig Maximilian University of Munich) and sponsored primarily by the startup Stability AI.
- The algorithm is based on ideas from *DALL-E 2* (developed by Open AI, which is also the creator of ChatGPT), *Imagen* (from Google), and others image generation models.
- Katherine Crowson, which works in Stability AI team, is one of the main responsibles for the “boom” regarding AI art in recent years. She was the first to combine VQGAN with OpenAI CLIP, and later developed the CLIP-guided diffusion method, which today underpins a large portion of AI-powered image generation applications and services.

Results obtained with VQGAN+CLIP. Despite being beautiful, many of the results were inconsistent. For this reason, it is more used to generate artistic images (mainly abstract).

It was released about one year before Stable Diffusion. It is an outstanding evolution in the open source available techniques for image generation task

STABLE DIFFUSION – SOME RESULTS

Prompt: Luke Skywalker ordering a burger and fries from the Death Star canteen.
Parameters: Steps: 50, Sampler: Euler a, CFG scale: 7.0, Seed: 2551426893, Size: 512x512

Prompt: Cute small cat sitting in a movie theater eating chicken wiggs watching a movie ,unreal engine, cozy indoor lighting, artstation, detailed, digital painting,cinematic,character design by mark ryden and pixar and hayao miyazaki, unreal 5, daz, hyperrealistic, octane render
Negative prompt: ugly, ugly arms, ugly hands,
Parameters: Steps: 25, Sampler: Euler a, CFG scale: 8.0, Seed: 3099373267, Size: 512x512

Prompt: Cute small dog sitting in a movie theater eating popcorn watching a movie ,unreal engine, cozy indoor lighting, artstation, detailed, digital painting,cinematic,character design by mark ryden and pixar and hayao miyazaki, unreal 5, daz, hyperrealistic, octane render
Negative prompt: ugly, ugly arms, ugly hands,
Parameters: Steps: 25, Sampler: Euler a, CFG scale: 8.0, Seed: 2470332296, Size: 512x512

Prompt: london luxurious interior living-room, light walls
Negative prompt: pink gold
Parameters: Steps: 20, Sampler: Euler a, CFG scale: 7.0, Seed: 668581889, Size: 512x512

More examples: [Stable Diffusion - AI artwork]

EXPERT ACADEMY

OTHER GENERATIVE AI MODELS

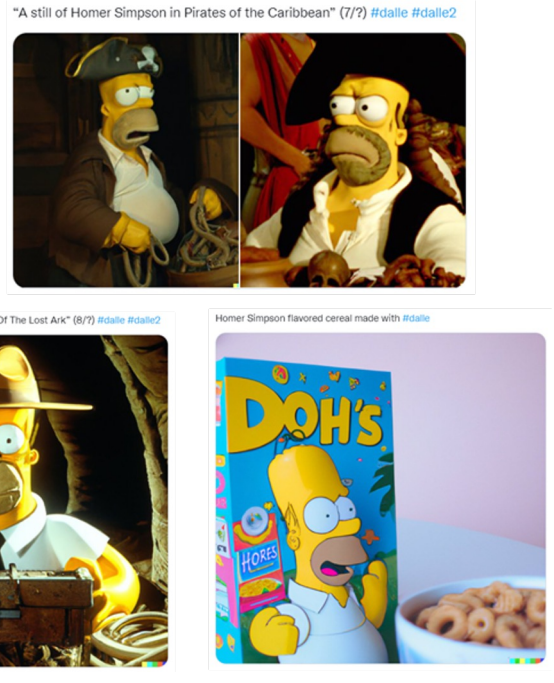
Midjourney

Examples generated with Midjourney v5

More examples: [Midjourney Community Showcase]

OTHER GENERATIVE AI MODELS

DALL-E



Generated with DALL-E 2

OTHER GENERATIVE AI MODELS

Imagen

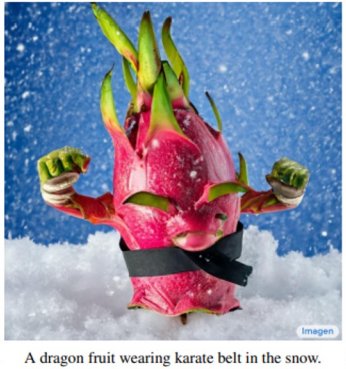
Sprouts in the shape of text 'Imagen' coming out of a fairytale book.

Teddy bears swimming at the Olympics 400m Butterfly event.

A cute corgi lives in a house made out of sushi.

A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

A brain riding a rocketship heading towards the moon.



A dragon fruit wearing karate belt in the snow.

A strawberry mug filled with white sesame seeds. The mug is floating in a dark chocolate sea.

Source: Imagen paper

OTHER GENERATIVE AI MODELS

GANs (Generative adversarial networks)

Although GANs are not focused on the text-to-image task, there are some GAN models that support this functionality.

Generated with VQGAN+CLIP

Generated with StyleGAN 2
(unconditional generation)

CycleGAN – for style transfer and domain change
(image-to-image)

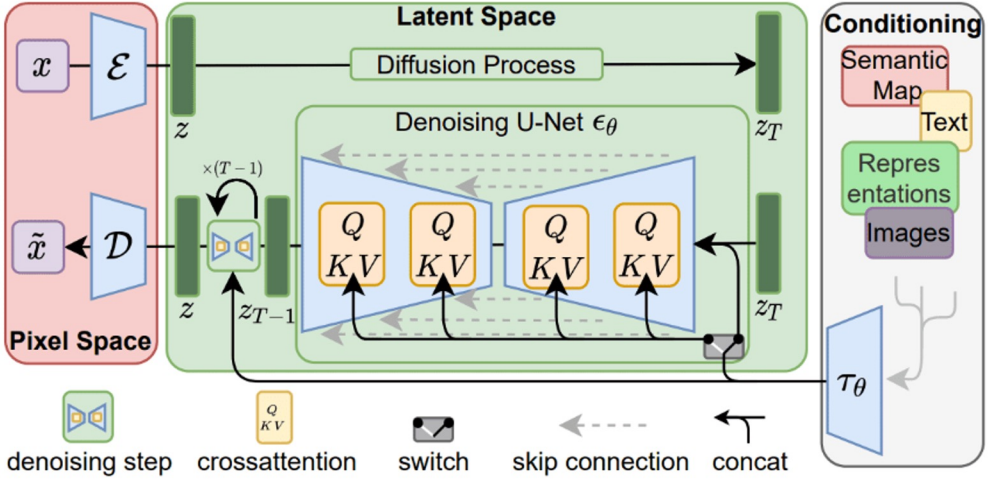
DIFFUSION MODELS

- Diffusion models are essentially Markov chains trained using variational inference.
- The purpose of diffusion models is to learn the latent structure of a dataset by modeling the way data points diffuse through the latent space.
- In the field of computer vision, this means that a neural network is trained to reduce noise from images blurred with Gaussian noise by learning how to reverse the diffusion process.
- In other words, it is the process that will transform noise (the initial state) into the generated image (the result).

LATENT DIFFUSION

Loss of a typical diffusion model $L_{DM} = \mathbb{E}_{\mathbf{x},t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2\right]$

Given an E encoder and a representation latent z, the loss for a latent diffusion model (LDM) is: $L_{LDM} = \mathbb{E}_{\mathcal{E}(\mathbf{x}),t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t)\|^2\right]$



Source: official paper https://arxiv.org/abs/2112.10752

STABLE DIFFUSION – HOW IT WORKS

Three main components:

1. **Autoencoder (VAE)**
2. **U-Net.**
3. **Text-encoder**

AUTOENCODERS

- **AutoEncoders** (AEs) - introduced as a powerful tool to compress images or data in general. It is composed of:
  - Encoder - converts the input into a lower-dimensional latent vector, usually through standard convolutions and clustering layers.
  - Decoder - performs operations such as deconvolution and upsampling to try to reconstruct the input image.
- These two CNNs are “controlled” with a Mean Squared Error (MSE) loss, which is simply the pixel-by-pixel difference between the original and the reconstructed image.

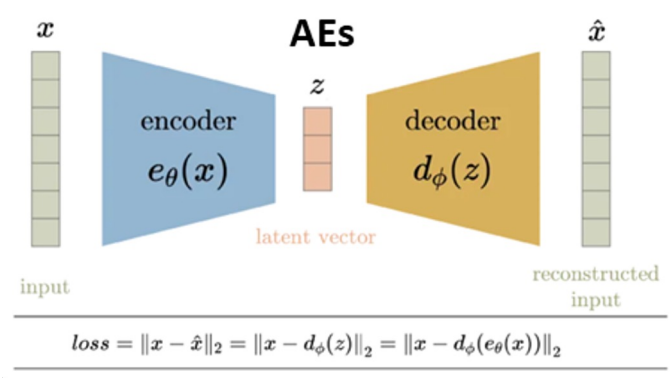


Image source: Towards Data Science

VARIATIONAL AUTOENCODERS

- The encoder returns the mean and the standard deviation for each input; then, the latent vector is sampled from this distribution and sent to the decoder to reconstruct the input.
- Training is now controlled by reconstruction loss and similarity loss, which is the KL divergence between the latent space distribution and the standard Gaussian.
- VAEs are trained not only to reconstruct images, but also to produce latent vectors from a normal distribution.

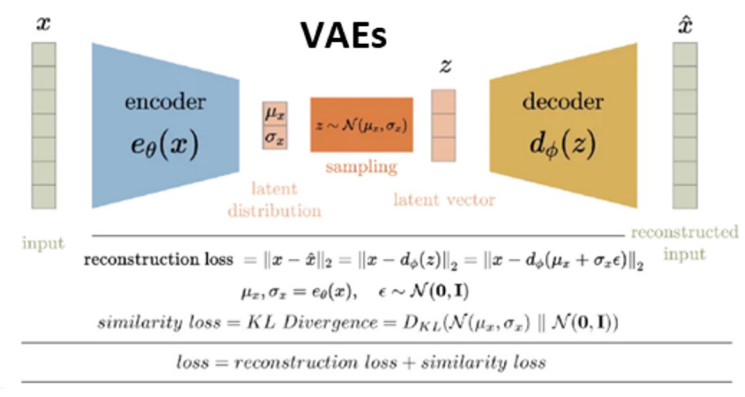
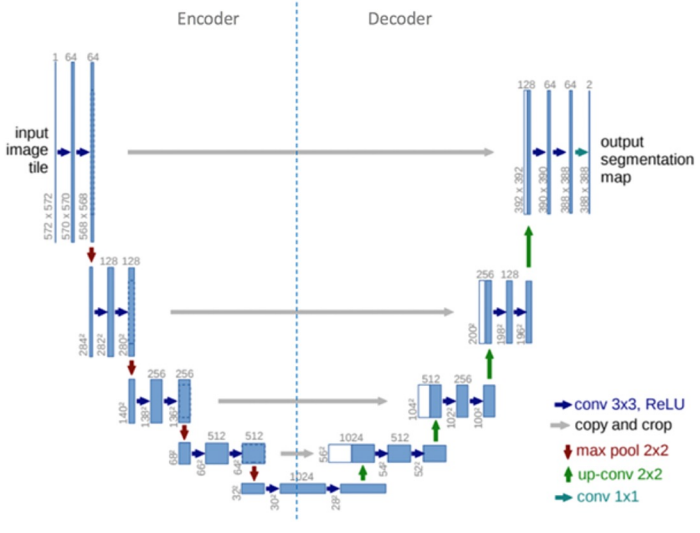


Image source: Towards Data Science

U-NET

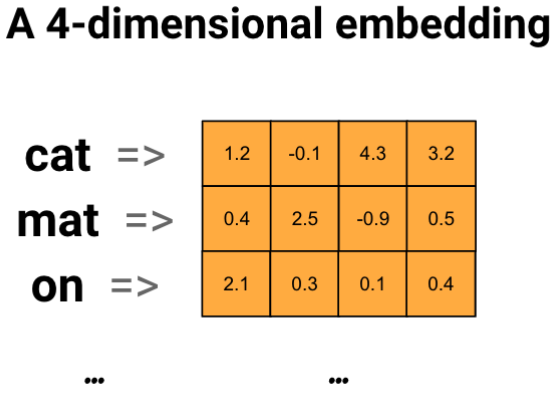
- **encoder** – it is the first part and it is usually a pre-trained classification network (like VGG/ResNet). Convolution blocks followed by maxpool/downsampling are applied to encode the input image into feature representations at several different levels.
- **decoder** – it is the second part of the architecture, which has the goal to semantically project the discriminative features (lower resolution) learned by the encoder into the pixel space (higher resolution) to obtain a dense classification. The decoder consists of upsampling and concatenation followed by regular convolution operations.



U-net original architecture.
Source: paper (https://arxiv.org/abs/1505.04597)

TEXT-ENCODER

- The text-encoder is responsible for turning the input prompt into an embedding space that can be understood by U-Net.
- It is usually a simple transformer-based encoder that maps a sequence of input tokens to a sequence of latent text-embeddings.
- Inspired by Imagen, Stable Diffusion does not train the text encoder during training, it simply uses an already trained CLIP text encoder, the **CLIPTextModel**.

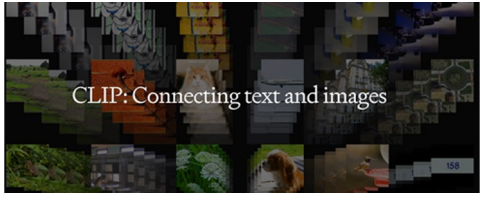


Source: https://www.tensorflow.org/text/guide/word_embeddings

CLIP

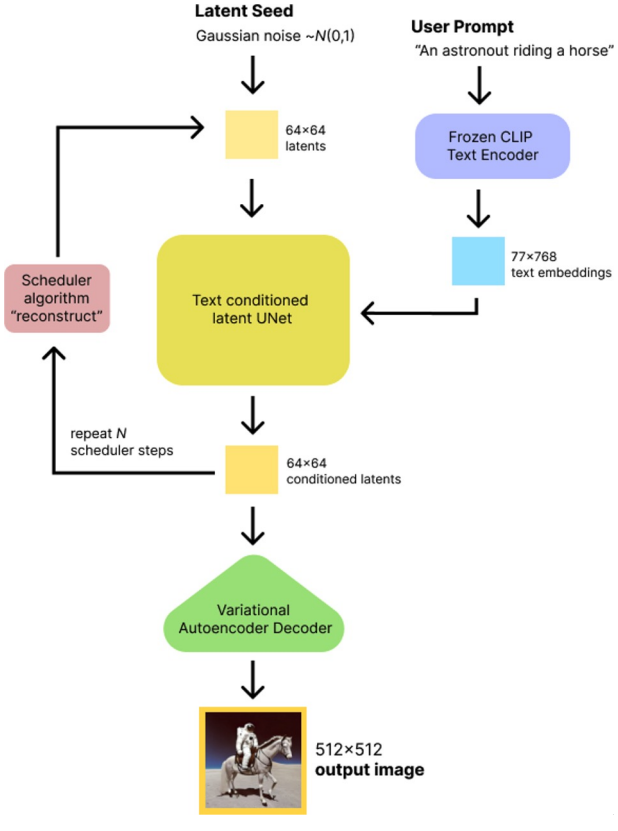
CLIP (Contrastive Language–Image Pre-training)

- Neural network developed by OpenAI.
- Efficiently learns visual natural language supervision concepts.
- More use cases (besides image generation) - image classification, image captioning, image similarity, content moderation, image search, etc.



STABLE DIFFUSION INFERENCE

- SD receives a latent seed and a text prompt as input.
- The seed is used to generate random representations of latent images of size 64x64.
- The text prompt is transformed into text embeddings of size 77x768 using CLIP text encoder.
- U-Net iteratively reduces noise from the random latent image representations while conditioning on the text embeddings.
- The U-Net output (noise residual) is used to compute a denoised latent image representation using a scheduler algorithm.
- The denoising process is repeated x times to recover the best latent image representations.
- Once completed, the latent image representation is decoded by the decoder part of the VAE.



IN-DEPTH EXPLANATION

The Illustrated Guide - Stable Diffusion

- https://jalammar.github.io/illustrated-stable-diffusion/

How diffusion models work: the math from scratch

- https://theaisummer.com/diffusion-models/

The authors describe step by step the more technical details about how the algorithm and how the model works.

STABLE DIFFUSION INFERENCE

Components:

- text_encoder: Stable Diffusion uses CLIP, but other diffusion models can use other encoders like BERT.
- tokenizer. It must match the one used by the text_encoder model.
- scheduler: The scheduler algorithm used to progressively add noise to the image during training.
- u-net: The model used to generate the latent representation of the input.
- vae: Autoencoder module that we will use to decode latent representations into real images.

SCHEDULER ALGORITHMS

- Scheduler algorithms (also called samplers) calculates the predicted denoised image representation from the previous noise representation and the predicted residual noise.
- Determines how the image is "calculated.
- There are several different algorithms. Some examples commonly used with Stable Diffusion:
    - PNDM (default)
    - DDIM Scheduler
    - K-LMS Scheduler
    - Euler Ancestral Discrete Scheduler *(Euler A)*
    - DPM Scheduler

For technical details about the theory and mathematics behind the algorithms, it is recommended to read the paper: *Elucidating the Design Space of Diffusion-Based Generative Models* - *https://arxiv.org/abs/2206.00364*

SCHEDULER ALGORITHMS - COMPARISON

Comparison of the same prompt using different schedulers (column: scheduler; row: value used for CFG parameter)

Source: artstation.com

STABLE DIFFUSION - IMPLEMENTATION

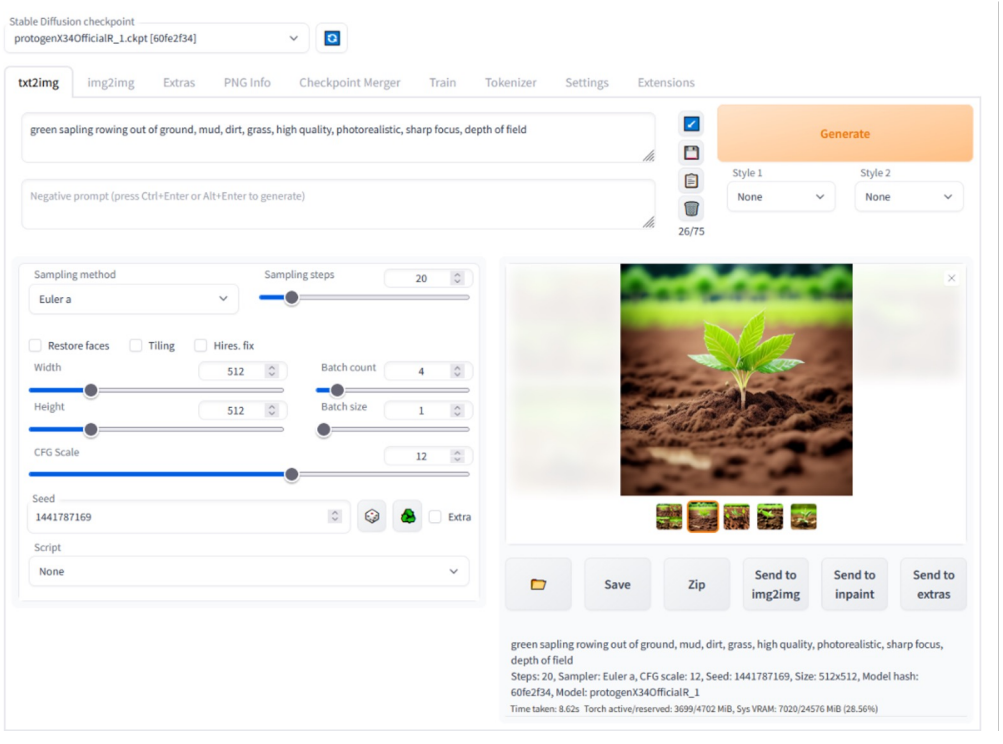
- The fastest way to test Stable Diffusion is through websites and online services such as:
  - https://www.mage.space/explore
  - https://huggingface.co/spaces/runwayml/stable-diffusion-v1-5
  - https://playgroundai.com/
- There are more advanced services that also provide other AIs and an easy interface for use them together, such as Leonardo.AI.
- There are several applications that are based on Stable Diffusion, new ones being released every week.
- The negative point of these options is that they can all be quite limited (requiring you to pay more to be able to generate more images) or are also limited in terms of some functionalities.

INTERFACES (UI) FOR STABLE DIFFUSION

Advantages

- Advanced prompting techniques.
- More convenience to manage models.
- Easy way of using Inpainting to fix or modify specific parts of an image, as the UI offers a brush tool to paint the image in real time.
- Among other features that can make the creation flow much more practical, especially for those who use more frequently or use it as a tool for work.

INTERFACES (UI) FOR STABLE DIFFUSION



A Web UI based on the Gradio library for Stable Diffusion made available by the repository https://github.com/AUTOMATIC1111/stable-diffusion-webui

INTERFACES (UI) FOR STABLE DIFFUSION

Popular and recommended web UI options for Stable Diffusion:

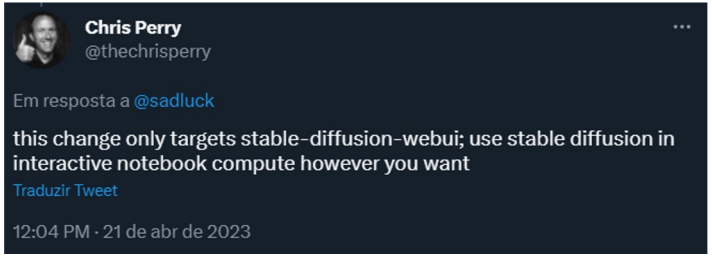
- https://github.com/AUTOMATIC1111/stable-diffusion-webui
- https://github.com/TheLastBen/fast-stable-diffusion
- https://github.com/camenduru/stable-diffusion-webui-colab
- https://stable-diffusion-ui.github.io

Note: these are the best solutions today, but this may change.

LIMITATION ON COLAB FREE PLAN

- The most known and perhaps most powerful WebUI tool is AUTOMATIC1111. On April 21, 2023, the use of this and others WebUIs were blocked in Colab.
- That same day, Colab changed the terms of use to not encourage the use of other remote Uis: https://research.google.com/colaboratory/faq.html#limitations-and-restrictions
- According to an official statement from Chris Perry - Product Lead at Google Colab - it is no longer possible to use these interfaces on the platform on the free tier (only on the paid plan, Colab Pro)

"Sorry. We prioritize interactive notebook compute for free of charge tier users - the webui is just using Colab as a convenient free GPU. We try to be cool, but usage has really been increasing lately and it's very noticeable now in our costs. I am constrained with what we can do with our budget!"



LIMITATION ON COLAB FREE PLAN

- However, some UI versions based on the Gradio library still seem to be accessible in Colab free plan, such as the https://github.com/TheLastBen/fast-stable-diffusion. After a few minutes the session is disconnected.

- Although they may be accessible for only a few minutes, it is possible that in the future their use will be completely limited also in Colab Pro, due to the changes in the terms of use.

- Meanwhile, you can use it at your own risk. The Product Lead has also confirmed that they don't plan to ban the user's accounts (they'll only display the warning message and automatically disconnect the runtime). So, there’s no risk on being banned for this cases - unless you misuses the technique and goes beyond ethical limits.

PROMPTS

- A **prompt** is a set of text instructions given to a deep learning algorithm to produce a defined output. These prompts are sent to the model to condition the image generation.
- One of the key aspects of AI image generation is that you need to master the technique of how to create effective text prompts.
- If the wrong prompts are used, the result will not be as you intended.
- The specialization in this technique is known today as P**rompt Engineering**.

BUILDING A GOOD PROMPT

- You need to exactly describe how you want the image (using a lot of details).
- Example: you want to generate an image of a man on the street.
- The AI will generate this to you, however, it will choose what the image will look like: it may not be the age group you would like, or the facial expression, location, pose, outfit, etc. It may not even generate a photography – so, if this is your goal, it's more guaranteed to also write it in the prompt (e.g.: photo, painting, etc.)
- If you're generating an image, you're definitely looking for a certain level of specificity, so please provide a good description of how you want that image.

BUILDING A GOOD PROMPT

The two main tips for building a good prompt are:

1. Describe the subject in detail.
2. Make sure to include powerful keywords to define the style you are looking for in the final image.

Being more specific, the anatomy of a good prompt should cover most of these features:

1. Subject / Object
2. Action and Local
3. Type
4. Style
5. Colors
6. Artist
7. Resolution
8. Website
9. Other features

BUILDING A GOOD PROMPT

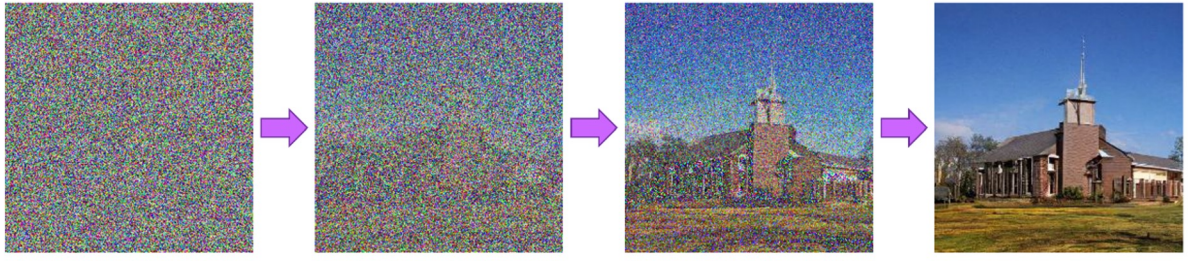
You can find a collection of prompts online and choose one that you like.

Some websites where you will find the prompts for professional images – and also more information, like the model and parameters used:

- https://lexica.art/
- https://civitai.com/
- https://www.mage.space/explore
- https://publicprompts.art/
- https://playgroundai.com/
- https://stablediffusion.fr/prompts
- https://github.com/Dalabad/stable-diffusion-prompt-templates

PARAMETERS - SEED

- The seed is a number used to initialize the generation of random numbers. It is used to create the initial latent noise.
- By setting a seed, we can reproduce the results.
- In the implementation, Stable Diffusion generates a random tensor in the latent space. You control this tensor by setting the random number generator seed. If you set the seed to a certain value you will always get the same random tensor.
- Remember that the AI is generating the whole image from noise - it's like taking several colored pencils together and painting the little dots in the sheet, until finally generating the desired image.



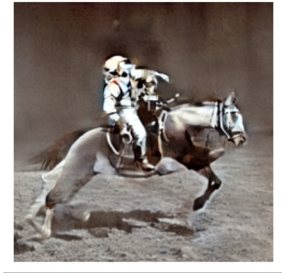
PARAMETERS - INFERENCE STEPS

The more steps, the better the results but the longer it takes to generate the image.

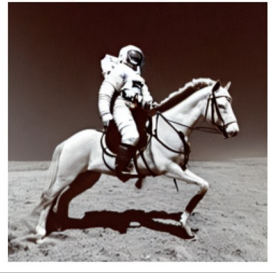
It is also known as **denoising steps**, as it indicates the number of steps required to turn the image from complete noise (initial state) into the result.

- Stable Diffusion works very well with a relatively small number of steps, so we recommend using the default value of 50 inference steps. The more steps the better the result, but there comes a point where the image stops improving.
- If you want faster results, you can use a smaller number. If you want potentially higher quality results, you can use higher numbers.
- The defaut number of steps varies according to the scheduler algorithm.

num_inference_steps: 10



num_inference_steps: 20



num_inference_steps: 30

num_inference_steps: 40

num_inference_steps: 50

PARAMETERS - CFG

The **classifier-free guidance** (CFG - also known as **guidance scale**) is a way to increase the adherence to the conditional prompt that guides the generation, as well as the overall quality of the image.

It controls how much the prompt will be taken into account for conditioning the diffusion process.

It represents how much importance will be given to the prompt when generating the images – the higher the value, the greater the importance.

- Smaller values: the more the prompt is ignored. For example, if the value is set to 0 then the image generation is unconditioned.
- Higher values: returns images that better represent the prompt.

NEGATIVE PROMPTS

- The negative prompt is an additional way of telling what you don't want in the image, what you'd like to avoid.
- It could be used to remove objects from the image or fix defects.
- It is optional in the first versions of Stable Diffusion, however, in the latest versions it has become really important to generate quality images.
- Some images can only be generated using negative prompts.

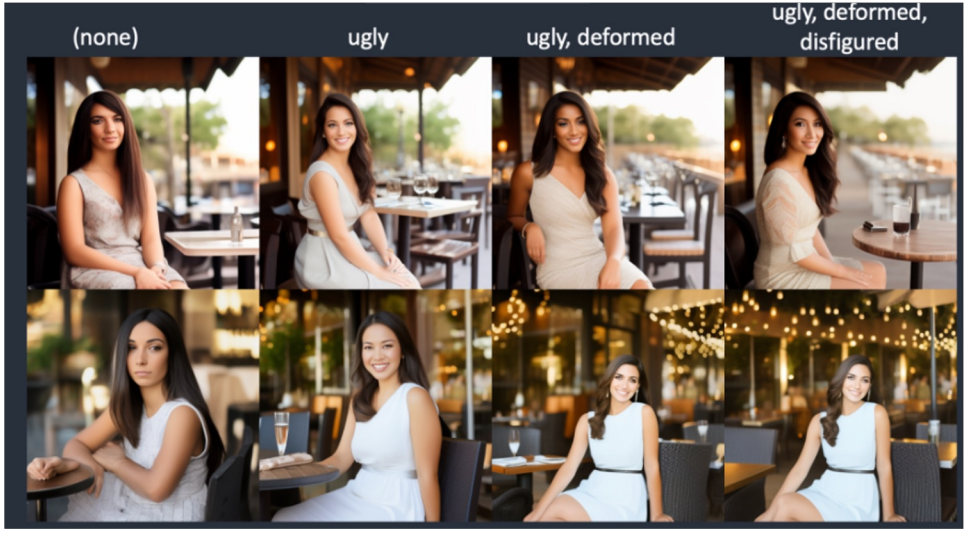


Image source: StableDiffusionArt.com

NEGATIVE PROMPTS

Let’s suppose we want to generate portraits of men. We then provide the prompt: *“Portrait photo of a man.”*

However, the network generated images of men with mustaches, and let's just say we don't want that. So, we change the prompt: *“Portrait photo of a man without mustache.”*

As a result we had even more prominent mustaches!

Why does it happen?

Stable Diffusion understood the prompt *“man”* and *“mustache”* as the “same”.

Image source: StableDiffusionArt.com

NEGATIVE PROMPTS

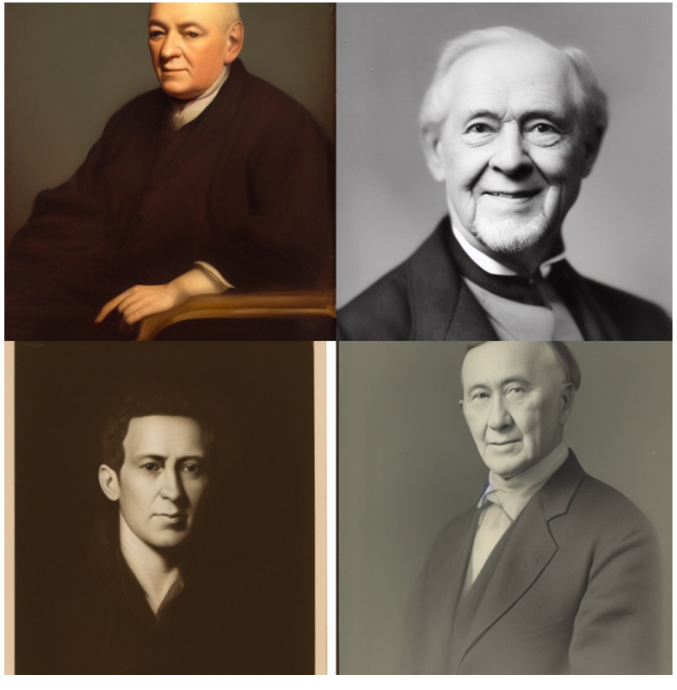
The negative prompt can be effective to solve the issue.

Let's try again to provide to the network the same prompt

```
> Portrait photo of a man
```

but now we also provide a negative prompt

```
> mustache
```

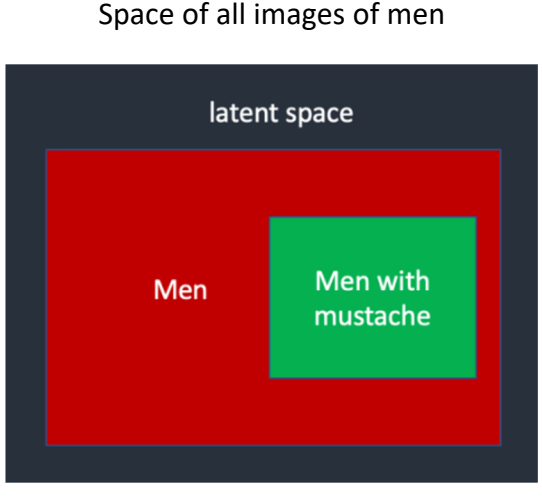


NEGATIVE PROMPTS

- During the text-to-image conditioning step, the prompt is converted into embedding vectors which will feed the U-Net noise predictor.
- There are two sets of embedding vectors: one for the positive prompt and one for the negative prompt.
- Both the positive and negative prompts have 77 tokens.

NEGATIVE PROMPTS

- When we use the prompt *“Portrait photo of a man”*, Stable Diffusion samples images of the entire latent space of all men, with and without a mustache. So, we get pictures of men with and without mustache.
- When the negative prompt *“mustache”* is added, the space *“men with mustache”* is excluded.
- Therefore, this way we are only sampling images of men without mustaches.



MODELS

- Stable Diffusion models (sometimes called as checkpoint files) are pre-trained weights used to generate images, for general or specific purposes.
- The images that a model can generate depend on the data used to train it. For example, a model will not be able to generate an image of a cat if there are no cats in the training data.
- If you train a model with images of cats, it should be able to only generate cats.
- The official models released by Stability AI and its partners are called base models, or general-purpose models.
- Some examples of base models: v1.4, v1.5, v2.0 e v2.1.

STABLE DIFFUSION V1

- **v1.4** - Released in August 2022 by Stability AI. It is considered the first Stable Diffusion model that was publicly available.
  - https://huggingface.co/CompVis/stable-diffusion-v-1-4-original

- **v1.5** - released in October 2022 by Stability AI and Runway ML. It is based on its predecessor, with additional training.
  - https://huggingface.co/runwayml/stable-diffusion-v1-5

Both are considered general-purpose models.

STABLE DIFFUSION V2

Differences in v2

- Stable Diffusion v1 uses CLIP Open AI for text embedding.
- Stable Diffusion v2 uses OpenClip for text embedding.

Main reasons for the change:

- OpenClip is currently five times bigger. A larger text encoder model improves image quality. It is able to better “understand” the prompt.

STABLE DIFFUSION V2

v2.0 - Released in November 2022 by Stability AI

In addition to 512 × 512 dimensions, a higher resolution version of 768 × 768 pixels is available.

v2.1 - Released in December 2022 by Stability AI

FINE-TUNING - METHODS

- **Additional training** – training a base model with an additional dataset. For example, you can train Stable Diffusion with an additional old car dataset to orient the aesthetics of the cars to that specific type.

- **Dreambooth** – initially developed by Google, it is a technique for injecting custom subjects into the models. Due to its architecture, it is possible to achieve great results using only 3/5 custom images.

- **Textual inversion** (also called *Embedding*) – it injects a custom subject into the model with just a few examples. A new keyword is created specifically for the new object and the training is executed only in the embedding neural network.

DREAMBOOTH

- **Dreambooth** is a method to fine-tune text-to-image models like Stable Diffusion.
- Published in 2022 by the Google Research team: *DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation*, by Ruiz et. al
- It works by injecting a custom concept into the existing model. This concept can be anything, for example: a person's face, an animal, an object, or even a specific style.
- This technique was developed to solve two problems when inserting the object into the model: overfitting (since the dataset is too small) and language drift.

Image source: https://dreambooth.github.io

DREAMBOOTH

Dreambooth solves these problems in two ways:

1. Using a rare word for the new subject, so it doesn't have too much meaning in the model.
2. Preservation of the class: to preserve the meaning of the class, the model is adjusted in such a way that the subject is injected while the generation of the class image is preserved.

The main advantage is that we need only few images (3~5) to get great results.

For example, to preserve the meaning of the class (dog, in the example below), the model is adjusted so that the subject (the specific dog, which will receive a unique identifier) is injected while the generation of the image of the class (dog) is preserved.

Image source: https://dreambooth.github.io

DREAMBOOTH - FINE-TUNING

Dreambooth training requires:

1. Unique identifier
2. Class name
3. Images of the subject to be inserted

In our implementation, we are going to use the face of a person. The unique identifier can be **zwx**, which needs to be a term associated with no concept or feature recognized by the model. The class is **person**.

We need to build the **instance prompt**

> a photo of [unique identifier] [class name] → e.g. *a photo of zwx person*

And a **class prompt**

> a photo of [class name] → e.g. *a photo of a person*

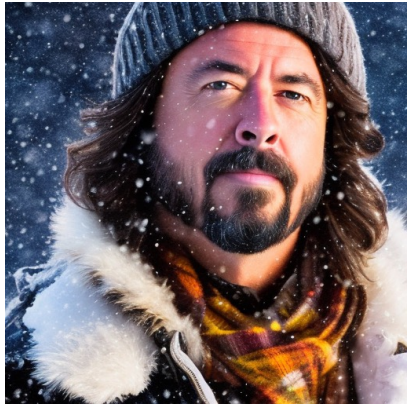
DREAMBOOTH - FINE-TUNING

Creating the training set

The quality of the dataset is essential for the proper functioning of the technique. To make the fine-tuned model more robust, it is preferable to avoid images that are all taken in the same environment, with the same lighting, angle, etc. To obtain good results, diversified images are needed.

DREAMBOOTH – FINE-TUNING RESULTS

photo of zwx person, in an armor, realistic, visible face, colored, detailed face, ultra detailed, natural lighting



digital painting of zwx in the snow, realistic, hd, vivid, sunset

photo of zwx person as an astronaut, natural lighting, frontal face, closeup, starry sky in the background

photo of zwx person in the forest, natural lighting, frontal face

DREAMBOOTH – FINE-TUNING RESULTS

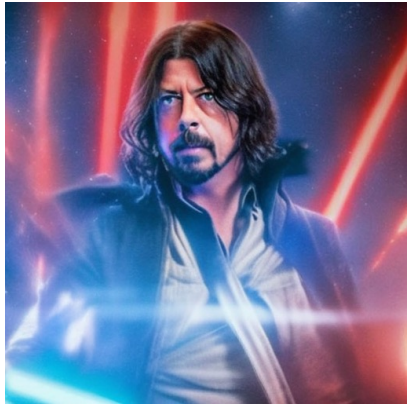


photo of zwx person in star wars, realistic, 4k ultra hd, blue and red tones

photo of zwx person in star wars, realistic

photo of zwx person in star wars holding a light saber, ultra realistic, 4k ultra hd, vivid colors, masterpiece, cinematic lighting

painting of zwx person in star wars, realistic, 4k ultra hd, blue and red tones

DREAMBOOTH – FINE-TUNING RESULTS

photo of zwx person in a western movie

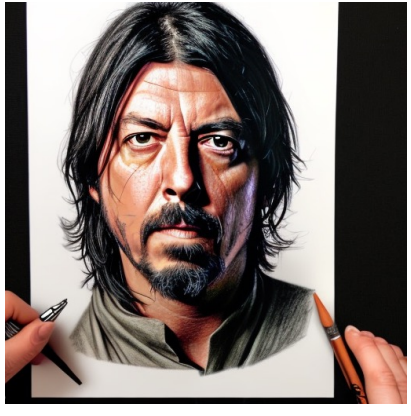
photo of zwx person in the desert, pyramids in the background

photo of zwx person in star wars holding a light saber, ultra realistic, 4k ultra hd, vivid colors, masterpiece, cinematic lighting

watercolor painting of zwx person, realistic, blue and orange tones

DREAMBOOTH – FINE-TUNING RESULTS

painting of zwx person in star wars, realistic, 4k ultra hd, blue and red tones



drawing of zwx person

photo of zwx person, wearing an iron man suit, realistic, portrait, closeup, natural lighting, hd

photo of zwx person in the forest, natural lighting, frontal face

IMPROVING DREAMBOOTH FINE-TUNING

> Common problems when testing the fine-tuned models

Case 1 - The images do not look like the trained subject.

- Make sure you are using the right prompt. For example: *[photo of zwx person]*. Putting the class name after the identifier might help.

Case 2 – The generated images look like the trained subject, but they are very similar to the training images.

- This can happen for a few reasons: you trained for longer than you should, or the training images are too similar, or you need to train with more images to get more diverse results.

IMPROVING DREAMBOOTH FINE-TUNING

> Common cases when testing the fine-tuned models

Case 3 - Images look like the subject, but not when tested in different styles.

- It probably means you haven't trained enough. You can train longer by changing the parameters or simply add more images.
- You could try to “reinforce” the information multiple times in the prompt.
- For example: *zwx person in a portrait photograph, zwx person in a medium format photo of zwx person*

IMPROVING DREAMBOOTH FINE-TUNING

> Common cases when testing the fine-tuned models

Case 4 - The generated images are noisy or the quality is bad.

- It probably means that overfitting has occurred (recap the previous recommendations).
- Make sure you are using the DDIM scheduler. Try also running more inference steps (+/-100 had good results in some experiments).

Case 5 – The results are bad even following all previous recommendations.

- We recommend testing other parameters because sometimes it is necessary to tune them to get the desired results. For example, changing the scheduler to "polynomial" instead of "constant".

IMPROVING DREAMBOOTH FINE-TUNING

- More training images and more steps is not necessarily better.
- Several experiments demonstrate that it is better to train with 30 images than 3000.
- Something between 10~30 training images is a very recommended amount.
- You could start with 10 (or even less), check the results and add more images.

For more details, experiments and comparisons: https://huggingface.co/blog/dreambooth

REFERENCES

- Stable Diffusion Paper - *High-Resolution Image Synthesis with Latent Diffusion Models* - https://arxiv.org/abs/2112.10752
- Dreambooth Paper - *DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation* - https://arxiv.org/pdf/2208.12242.pdf
- Imagen Paper - https://arxiv.org/pdf/2205.11487.pdf
- Classifier-Free Diffusion Guidance Paper https://arxiv.org/pdf/2207.12598.pdf

- https://blog.paperspace.com/generating-images-with-stable-diffusion/
- https://huggingface.co/docs/diffusers/training/dreambooth
- https://stable-diffusion-art.com/prompt-guide/
- https://huggingface.co/docs/diffusers/using-diffusers/img2img
- https://research.runwayml.com/publications/high-resolution-image-synthesis-with-latent-diffusion-models