Stable Diffusion, Summarized

Taking a Look at how Diffusers Dream

DIFFUSION CREATING MODELS

A concise, high level overview on the mechanisms of stable diffusion.

AUTHOR
Salman Naqvi

PUBLISHED
Thursday, 13 April 2023

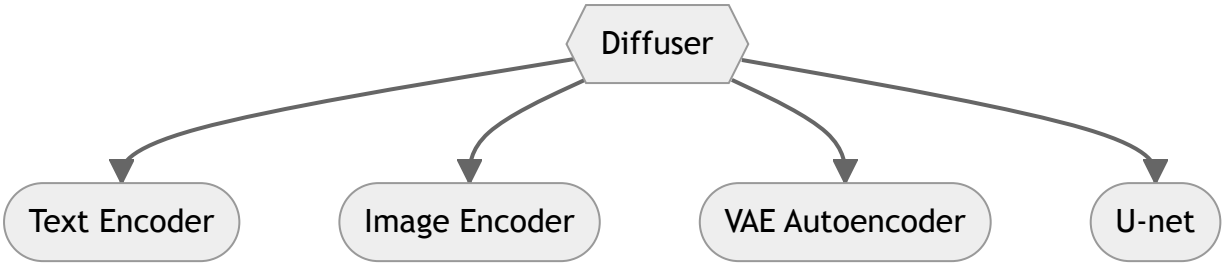
This post was edited on Sunday, 30 April 2023

Here, I explain the workings of stable diffusion at a high level.

Components

A diffuser contains four main components

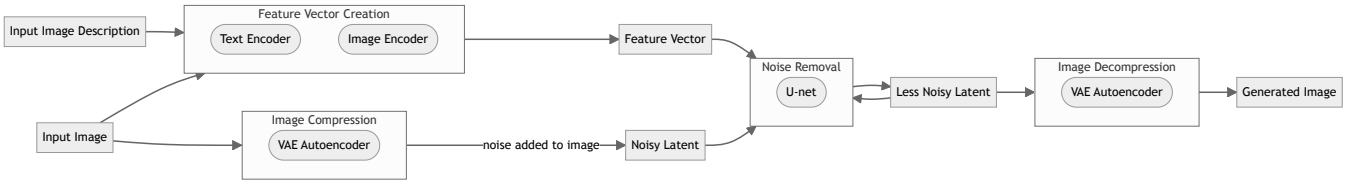
- The text encoder
- The image encoder
- The autoencoder (VAE autoencoder)
- The neural network (U-net)



Training

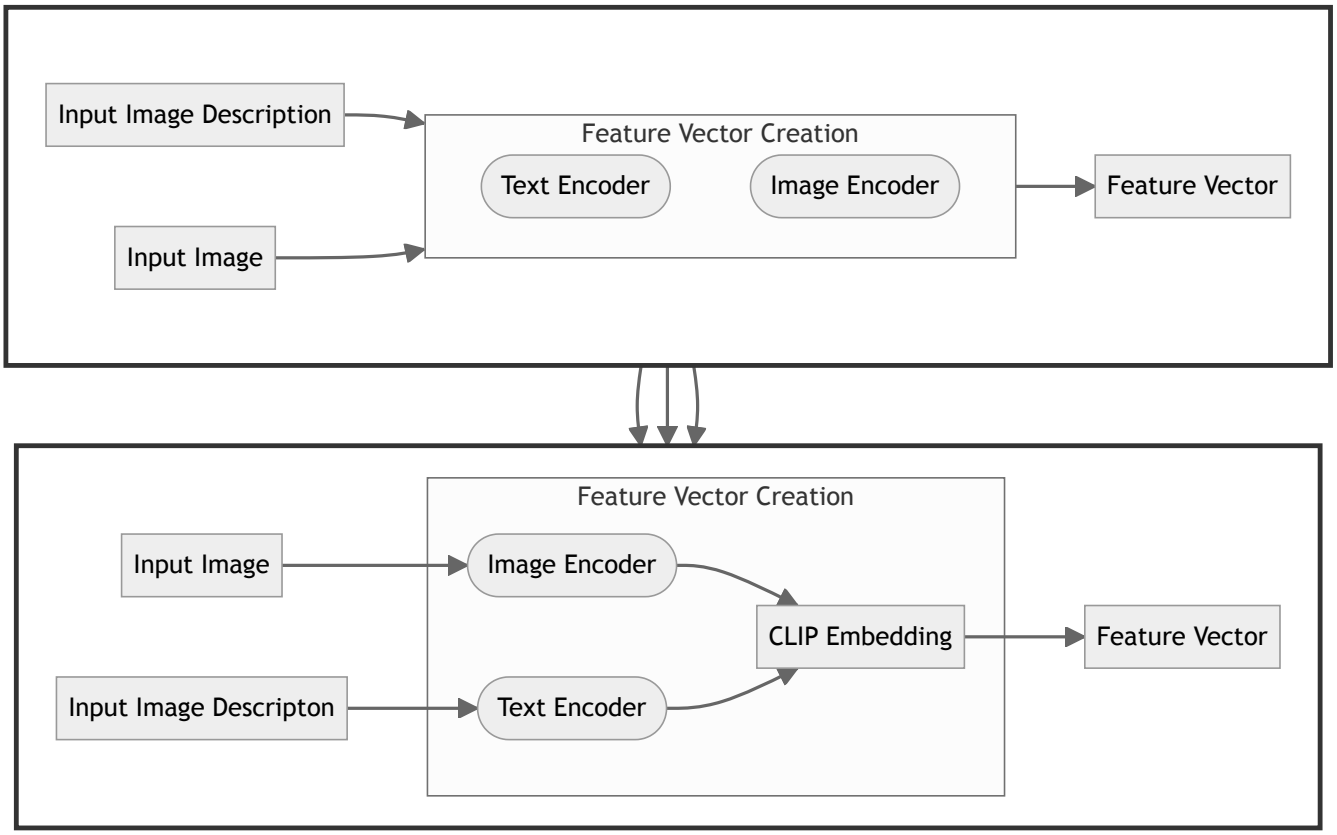
I'll explain the training process in terms of a single image.

When all components shown above are put into their respective places, the overall training process looks like this.



Let's break it down.

Feature Vector Creation

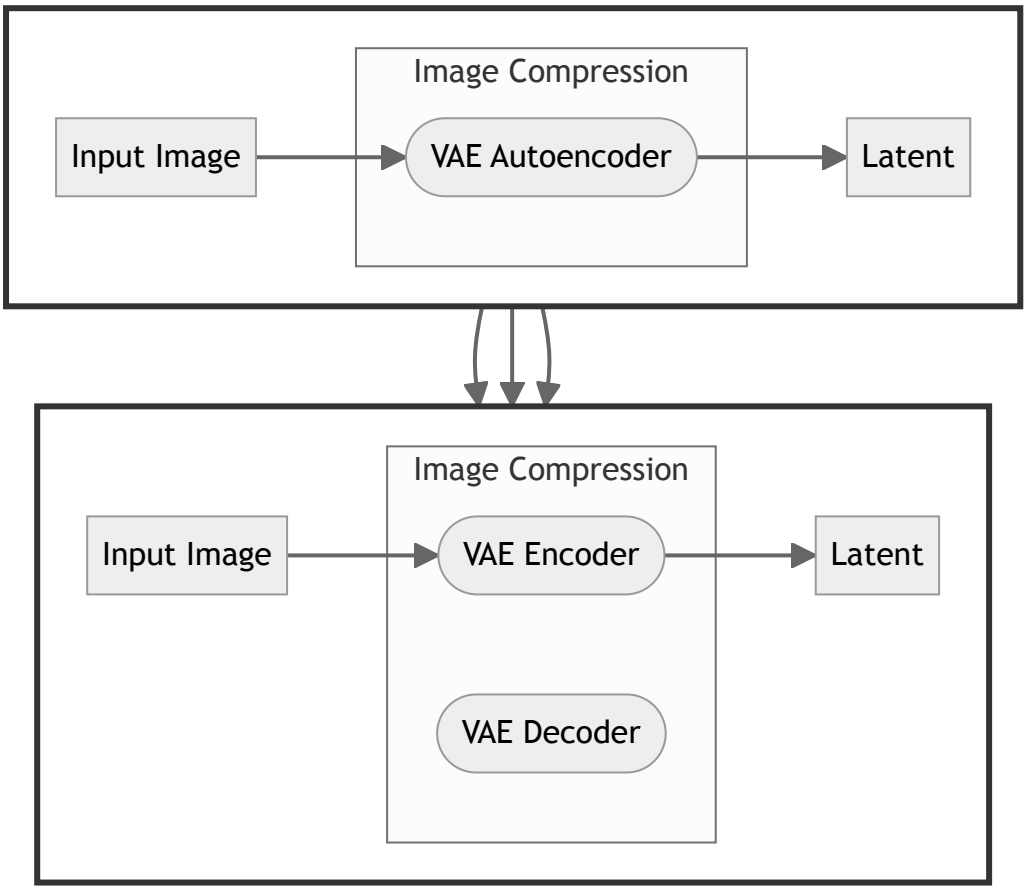


We start with an image and its description. The image encoder takes the image and produces a feature vector — a vector with numerical values that describe the image in some way. The text

encoder takes the image's description and similarly produces a feature vector.
These two feature vectors are then stored in what's known as a CLIP embedding. An embedding is simply a table where each row is an item and each column describes the items in some way. In this case, the rows represent feature vectors, and the columns are each feature in the vector.

Both encoders keep producing feature vectors until they are as similar as possible.

Image Compression

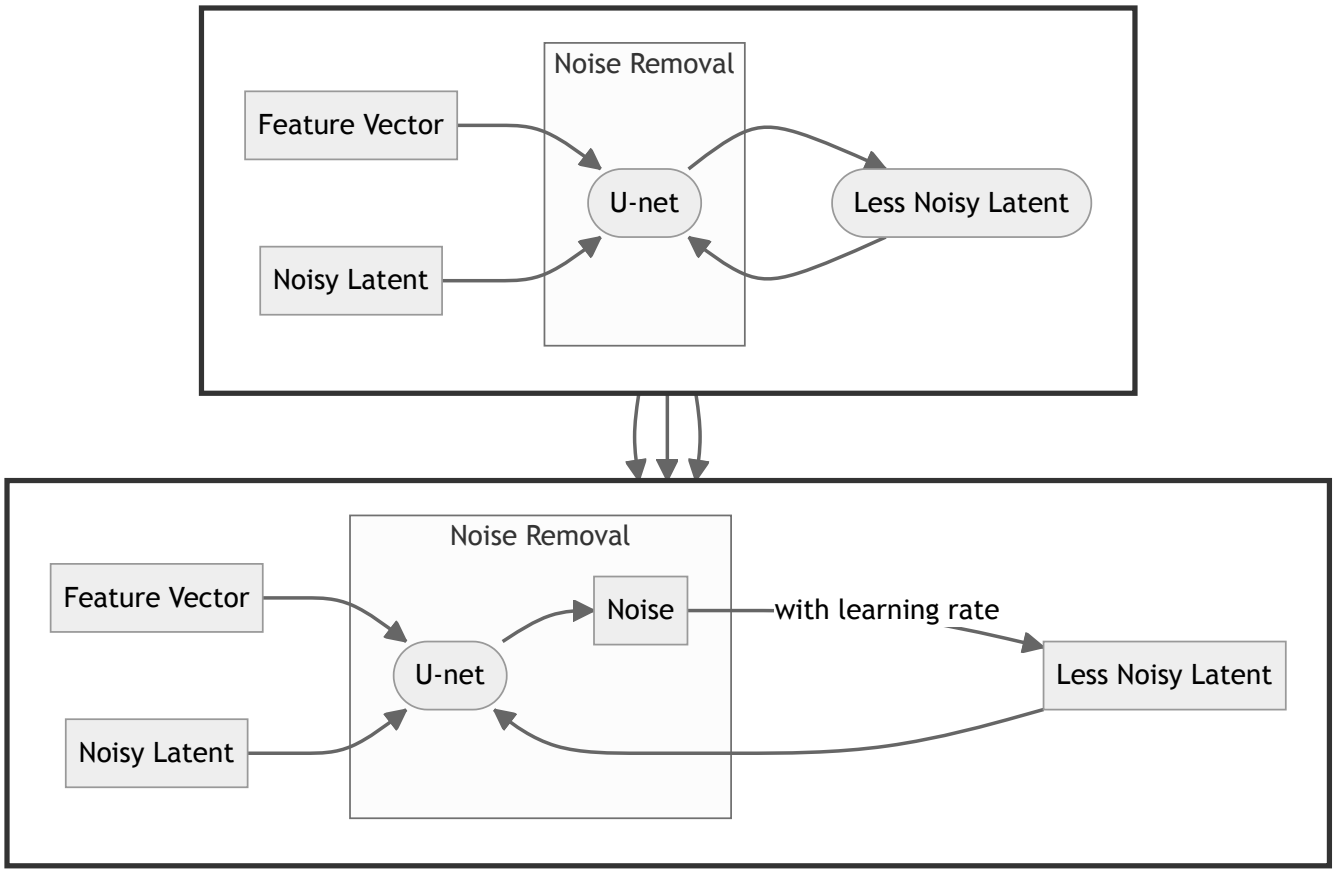


Once the feature vectors have been produced, the image is compressed by the VAE autoencoder. Some noise is then tossed onto the image.

The VAE autoencoder contains an encoder and a decoder. The encoder handles compression whereas the decoder handles decompression.

The compressed noisy image is now known as the latent. The image is compressed for faster computation, as there would be fewer pixels to compute on.

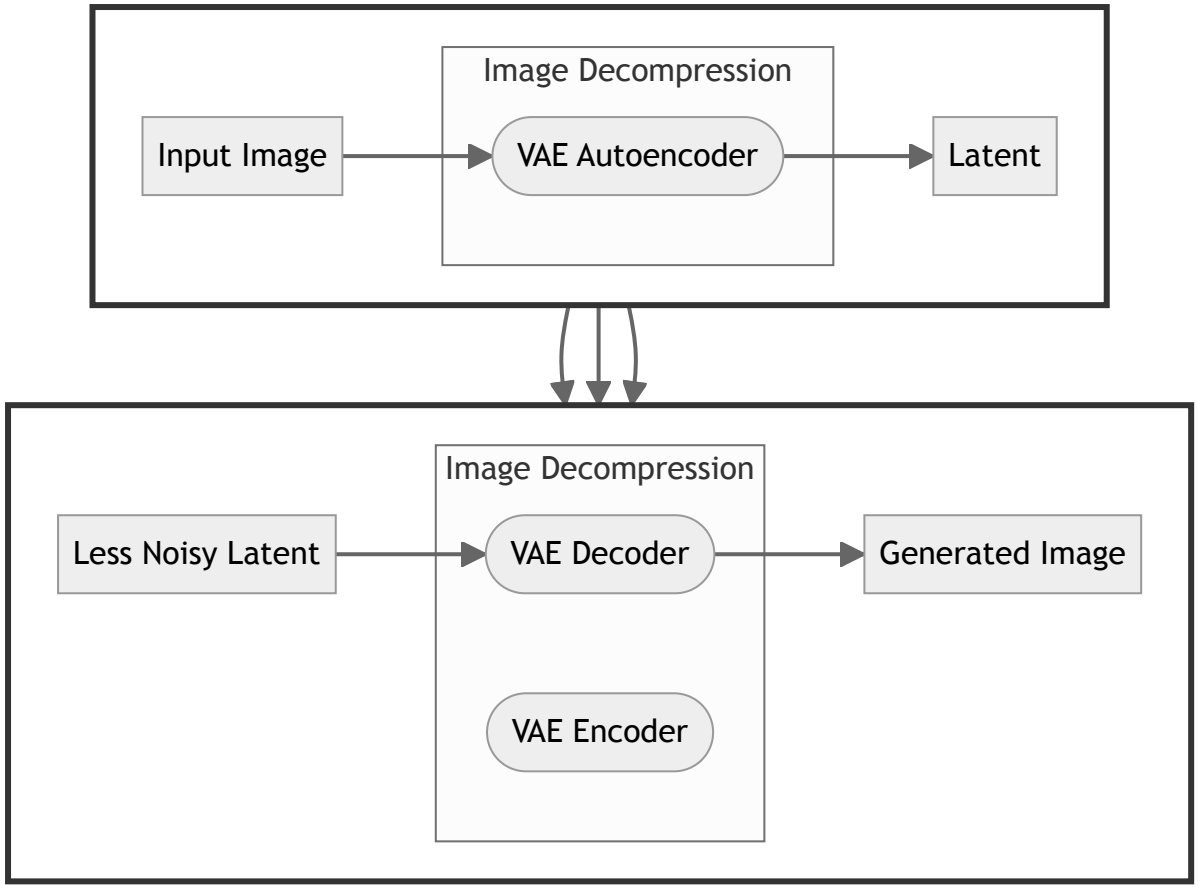
Noise Removal



The latent, together with its feature vector, is now input to the U-net. Instead of predicting what the original, un-noisy image was, the U-net predicts the noise that was tossed onto the image.

Once it outputs the predicted noise, that noise is subtracted from the latent in conjunction with the learning rate. This new, less noisy latent is now input again and the process repeats until desired.

Image Decompression



The latent is now decompressed through the VAE autoencoder's decoder.

We now have a generated image!

Inference

When using a diffuser for inference, the diffuser *typically* begins with a purely noisey latent. The diffuser uses the input prompt to guide the removal of noise from the latent, until the latent resembles what is desired.

Conclusion

And that's all there is to it!

We take an image and its prompt, and create a feature vector out of them. The image is compressed and noise is then added to it. The latent and the feature vector are input to a U-net which then predicts the noise in the latent. The predicted noise is subtracted from the latent, which is then input back to the U-net. After the desired number of steps has lapsed, the latent is decompressed and the generated image is ready!

If you have any comments, questions, suggestions, feedback, criticisms, or corrections, please do post them down in the comment section below!

0 reactions

2 comments · 2 replies – *powered by giscus*

Oldest Newest