Capstone Final Report

Text to Image synthesis by Stable Diffusion Model

Zhengyuan Zhang

11/25/2023

1 Introduction

In the evolving landscape of generative models and advanced image processing, Stable Diffusion emerges as a groundbreaking approach, particularly notable for synthesizing detailed images from textual descriptions. This technique, which ingeniously leverages deep learning, is at the forefront of transforming and manipulating images in unprecedented ways. Its essence lies in the controlled alteration of images, balancing preserving the original structure and integrating new changes. Powered by sophisticated neural architectures, such as CLIP for nuanced text understanding and Denoising Diffusion Probabilistic Models for image generation, Stable Diffusion can create striking, high-quality visuals from mere text prompts. This versatility extends across various applications, from artistic creation and data augmentation to enhancing image resolution in fields like medical imaging. Significantly, the process is inherently controllable, allowing precise tailoring of the changes applied to an image. This project aims to construct a Stable Diffusion model from scratch, exploring its potential in various text-to-image and image-to-image applications.

2 Related Work

The development of my Stable Diffusion model, built from scratch, is deeply rooted in the exploration and implementation of various pioneering works in the field of generative models and image processing. [1] provides the overarching framework and principles of the Stable Diffusion model. To enhance the model's image processing capabilities, algorithms from Ho et al.'s work [2] provide a framework for the gradual refinement and transformation of images through denoising steps. This process is intricately supported by incorporating the U-Net architecture, as initially detailed in Ronneberger et al.'s study [3]. In the Stable Diffusion model, U-Net is employed as the noise predictor. Additionally, the study by Kingma and Welling [4] offers critical insights into the implementation of variational autoencoders (VAEs), which are an integral part of achieving efficient latent space representation. Furthermore, integrating the CLIP model, as outlined in Radford et al.'s work [5], allows for the effective synthesis of images from textual inputs. The amalgamation of these diverse yet interconnected research works has laid a robust foundation for my project.

3 Methods

Since this project is focused on building the Stable Diffusion model from scratch, this section will outline the comprehensive process undertaken for its development. This approach involves a detailed, step-by-step elucidation of each key component, including Variational Autoencoders (VAEs), U-net architecture, and other integral elements. The focus will be on developing, integrating, and optimizing these components, providing insight into their individual functionalities and how they synergize within the overarching model structure. Subsequent subsections will delve deeper into each component, offering a detailed view of their roles and interactions within the

model's structure. [1]

3.1 VAEs

Encoder The encoder is used for transforming input images into a latent space representation. It begins with a convolutional layer that alters the channel dimensions while maintaining the input's height and width, preparing the image for deeper processing. Integral to the encoder's architecture are multiple residual blocks that play a critical role in preserving important information from the input and enabling the network to capture complex features. These blocks are crucial for maintaining the integrity of the image during the encoding process.

To process the input further, the encoder employs additional convolutional layers that downsample the image, reducing its spatial dimensions and increasing the depth of the feature maps, thus focusing on higher-level features. The encoder concludes its process with a convolutional layer that prepares the output, which is then split into two key components: mean and log variance. These components are essential for the reparameterization trick in Variational Autoencoders, introducing necessary variability into the latent representations. The log variance is clamped within a specific numerical range to ensure stability. From it, the standard deviation is derived.

The reparameterization process involves adding noise, scaled by this standard deviation, to the mean. Lastly, the output is scaled by a specific constant factor, ensuring a consistent and stable representation in the latent space.

Decoder The decoder functions to reconstruct images from their latent space representation. It begins by employing attention mechanisms, focusing on specific aspects of the encoded data to enhance details and fidelity in the output. Integral to its architecture are residual blocks that preserve important information, ensuring the integrity of the image is maintained during reconstruction.

[1] This section provides a general explanation of the code used to build the stable diffusion model.

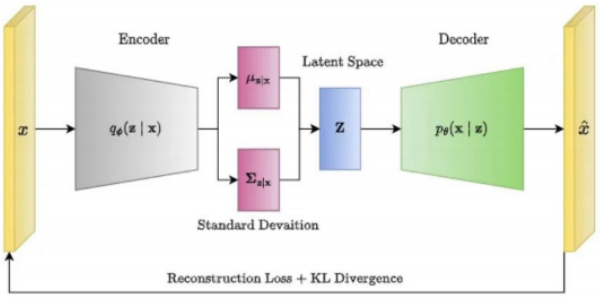

Figure 1: Architecture of VAEs model

The decoder also utilizes upsampling processes to increase the spatial dimensions of the image, a crucial step for accurately capturing and depicting fine details. Group normalization is applied throughout to stabilize the data across various layers, a key factor for effective processing and consistent quality of the features. The culmination of the decoding process involves transforming the upscaled and refined features back into a coherent final image. This includes specific adjustments to negate any scaling effects from the encoding process, with each step applied sequentially to the input. The result is a detailed, high-fidelity image.

3.2 CLIP

The CLIP is crucial in processing textual data for image synthesis tasks. It begins by converting text tokens into dense vector representations, transforming discrete textual elements into a format the neural network can process. This is followed by the incorporation of positional information, encoding the sequence and structure of the text, which is essential for understanding the context and relationships within sentences. The core functionality of the CLIP model includes a series of normalization and self-attention mechanisms, allowing the model to focus on different parts of the textual input in a context-aware manner. This is particularly important for tasks requiring a nuanced understanding of text and images. The process concludes with a feedforward layer that further processes the data, ensuring the textual information

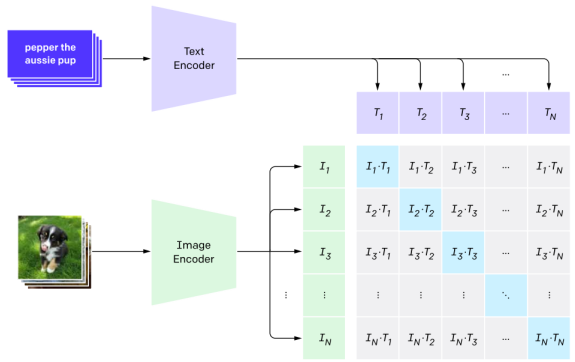

Figure 2: Architecture of CLIP

is comprehensively encoded. Sequentially processing tokens through these embedding and attention structures results in a final output accurately representing the textual description.

3.3 U-Net (Noise Predictor)

The U-Net functions as the noise predictor. It starts by receiving a latent representation of noise as its input and predicting the amount of noise to subtract from this latent representation, thereby gradually aligning it with the target prompt. The process begins with a time embedding module, which is instrumental in converting temporal input into a complex, higher-dimensional space. This conversion is crucial for embedding time dynamics into the generative process, ensuring that the temporal aspects are appropriately represented in the evolving image. At the core of the U-Net are several advanced processing blocks designed to handle image features at varying scales and depths. These blocks, composed of group normalization and convolutional layers, excel at integrating temporal information with spatial features. They achieve this through a linear transformation of the time embedding, effectively marrying the temporal dynamics with the spatial characteristics of the image.

The presence of residual connections in these blocks is key to preserving the integrity of the image's features during complex transformations, also simplifying the training of the deep network. Additionally, the U-Net architecture incorporates adjustments in channel dimensions and employs nonlinear activations (SiLU). These elements are essential for capturing complex, nonlinear relationships within the data. The output of the U-Net, still in the form of a latent representation, has been adjusted in terms of noise content, making it primed for the subsequent stages of the diffusion process.

3.4 Sampler

The sampler controls the diffusion process, iteratively subtracting noise from the noisy latent representation to recover a cleaner image that aligns with the target prompt. It starts by setting up a schedule of noise levels, known as betas, which are integral to the diffusion process. These betas are derived from the Gaussian noise schedule defined in Eq.(1) and Eq.(2)[2],

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}),$$
$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \quad (1)$$
$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I) \quad (2)$$

which guide the gradual addition of noise to the data, ensuring that the latent variables evolve following a Markov chain. The cumulative product of the complementary alphas, as seen in Eq.(2), determines the precise noise level at each diffusion step.

The reverse process is at the heart of the sampler's functionality, where noise is methodically subtracted from the noisy latent representation based on the betas. This subtraction is precisely controlled by the cumulative product of the complementary alphas, allowing the sampler to ascertain the noise level at each diffusion step accurately. The reverse process relies on conditioning the subsequent state on both the initial data point x_0 and the current state, informed by the theoretical underpinnings of Eq.(3) and Eq.(4). The sampler's role is to compute the predicted original sample by adjusting the mean and variance according to these equations, which enables the precise

[2]All the equations are from [2]

reversal of the noise trajectory.

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I), \quad (3)$$

$$\text{where } \tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha}_t(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t,$$
$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \quad (4)$$

The operational sequence within the sampler involves executing a diffusion step, wherein the estimated original sample is combined with the current sample. This combination is executed without the addition of noise, contrary to the forward process. Instead, noise is subtracted as dictated by the reverse process equations. Subsequently, the denoised latent representation is fed back into the noise predictor—implemented as a U-Net—for additional denoising assessment. Through a series of such iterative interactions, the sampler fulfills its purpose in the reverse diffusion process, effectively reducing noise levels and refining the image.

3.5 Combination

The following paragraph describes the file I wrote for combining all the components of the Stable Diffusion mentioned in the previous section. So that the model can achieve the text-to-image generation. It starts by setting the necessary dimensions for the output images and their corresponding latent representations. Key to this implementation is the handling of prompts. When enabled, Classifier Free Guidance (CFG) subtly enhances the process, especially for conditional prompts. This technique is employed after conditioning the model with the textual prompt, adjusting the output by a weight factor that reflects the importance of the conditioning signal. The CFG formula,

$$\text{output} = w \cdot (\text{output}_{\text{conditioned}} - \text{output}_{\text{unconditioned}}) + \text{output}_{\text{unconditioned}} \quad (5)$$

is applied to merge the conditioned and unconditioned outputs, with w acting as a scaling factor to control the influence of the prompt. In practice, this means that for each iteration of the diffusion process, the model's output is adjusted to ensure the generated image retains a strong link to the conditioning prompt, thereby enhancing the relevance and clarity of the generated content in relation to the text prompt provided.

The noise management component, fundamental to the diffusion model, is configured with specific parameters to control the refinement of the latent representations. This involves manipulating noise levels in a way that gradually enhances the quality of the generated image. In cases where an input image is provided, this image is first standardized and transformed into a format suitable for processing. Without an input image, a randomized latent representation is generated instead.

The image generation unfolds through an iterative process, progressively refining the latent representations. This refinement is underpinned by time-based and contextual guidance, ensuring that each iteration brings the image closer to the desired output. The progress of this refinement is meticulously tracked, with each step aimed at methodically reducing noise and denoising the image.

The transformed latent representations are converted back into image space following this iterative refinement. The resultant images undergo final adjustments to ensure they are in a suitable format for output. Supplementary functions facilitate key data transformations throughout this process, ensuring the images are visually aligned with the prompts and maintain a high-quality standard.

3.6 Model Weights Upload

To ensure the stable diffusion model generates high-quality images, training it on a large and diverse dataset of text-image pairs is critical. However, device limitations often make training such large models from scratch impractical. This project utilized model weights from pre-trained Stable Diffusion and CLIP models to circumvent this. Specifically, the model weights incorporated are from [6]. Leveraging these pre-trained weights allows the variational autoencoders (VAEs), CLIP, and U-Net within the project to inherit the learned patterns and knowledge

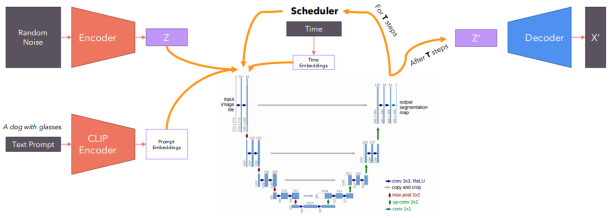

Figure 3: Architecture of Text to Image model

embedded in the pre-trained model. This transfer of learning ensures that the models within the project are initialized with weights that have already been optimized on vast datasets, equipping them with the ability to produce high-quality images without the need for extensive computational resources for training.

4 Results

In this section, I aim to demonstrate the model's capabilities in two distinct tasks: text-to-image generation and image-to-image translation. These tasks will showcase the model's proficiency in understanding and interpreting textual descriptions to create coherent and relevant visual outputs and its ability to transform an existing image in line with a given text prompt.

4.1 Text-to-Image

The architecture of the text-to-image process is shown in **Figure 3**. The process initiates with selecting a prompt that sets the creative direction. This prompt leads to the generation of random noise that Variational Autoencoders encode to form a latent representation. This representation is then introduced to the U-Net, which receives a conditioning signal corresponding to the prompt. The U-Net evaluates the noise and iteratively works with the sampler to denoise this latent space. This loop continues until the noise is sufficiently removed. Finally, the output is passed through a decoder, producing a final image that accurately reflects the initial textual prompt. **Figure 4** displays a selection of images produced by the model, showcasing their high quality and strong alignment with the provided text descriptions.

(a) The Eiffel Tower in the style of sketch

(b) Two people dancing in the snow

(c) A plane on fire in the air

(d) A zombie in the style of Picasso

Figure 4: Text-to-Image generation

4.2 Image-to-Image

Figure 5 illustrates the Image-to-Image transformation process, which closely mirrors the architecture of the Text-to-Image process. The key distinction lies in the starting point: an initial image is used as input. This process adapts the input image to align with a given text prompt. It begins by encoding the input image to obtain its latent representation. Subsequently, noise is introduced to this latent space. The degree of noise added grants the U-Net varying levels of freedom to modify the image. **Figure 6** presents a series of images that have been altered by the model, demonstrating the effectiveness of this process

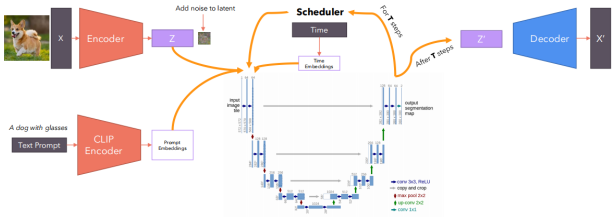

Figure 5: Architecture of Image to Image model

(a) Initial Image: a golden retriever

(b) A dog with glasses

(c) Initial Image: a forest

(d) The forest in winter

Figure 6: Image-to-Image generation

5 Conclusion

Text-to-Image (T2I) technology has reached a level of maturity within the field of machine learning, characterized by an array of advanced models adept at producing detailed and intricate images. Building upon this foundation, my future focus will be on extending the capabilities of Stable Diffusion to encompass Text-to-Video tasks. This expansion aims to harness the potential of Stable Diffusion in generating dynamic, video-based content from textual descriptions.

References

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[5] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[6] RunwayML, "Stable diffusion v1-5," 2023.