ISSN 2278-3091
Volume 12, No.2, March - April 2023
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse081222023.pdf
https://doi.org/10.30534/ijatcse/2023/081222023

Novel Implementation of TEXT2IMAGE

Dr. Imtiyaz Khan[1], Mohammed Junaid Adil[2], Mohammed Ahmed Hussain[3], Mohammed Sahil Arman[4]
[1]Assisatant Professor CS&AI Dept, MJCET OU, India, imtiyaz.khan@mjcollege.ac.in
[2]**B.E.CSE AI DS MJCET OU**, India, mdjunaidadil@gmail.com
[3]**B.E.CSE AI DS MJCET OU**, India, ahmedhussain1777@gmail.com
[4]**B.E.CSE AI DS MJCET OU**, India, sahilyfsm@gmail.com

Received Date February 23, 2023 Accepted Date: March 29, 2023 Published Date: April 06, 2023

ABSTRACT

Text-to-image generation has traditionally focused on finding better modelling assumptions for training on a fixed dataset. These assumptions might involve complex architectures, auxiliary losses, or side information such as object part labels or segmentation masks supplied during training. By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond. These models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders. Training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation. Latent diffusion models (LDMs) achieve new state-of-the-art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including text-to-image synthesis, while significantly reducing computational requirements compared to pixel-based DMs.

Key words : Text-to-image, diffusion models, latent diffusion models, synthesis.

1. INTRODUCTION

A text-to-image model is a machine learning model which takes as input a natural language description and produces an image matching that description. Such models began to be developed in the mid-2010s, as a result of advances in deep neural networks[2,6]. In 2022, the output of state-of-the-art text-to-image models, such as OpenAI's DALL-E 2. The most effective models have generally been trained on massive amounts of image and text data scraped from the web. Stable Diffusion is a latent text-to-image diffusion model capable of generating photo-realistic images given any text input[1]. It is a Latent Diffusion Model that uses a fixed, pretrained text encoder (CLIP ViT-L/14)[5].

2. LITERATURE SURVEY

Text-to-image generation has traditionally focused on finding better modelling assumptions for training on a fixed dataset. These assumptions might involve complex architectures, auxiliary losses, or side information such as object part labels or segmentation masks supplied during training. We describe a simple approach for this task based on a transformer that autoregressively models the text and image tokens as a single stream of data. With sufficient data and scale, our approach is competitive with previous domain-specific models when evaluated in a zero-shot fashion[2].
By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond[1,2]. Additionally, their formulation allows for a guiding mechanism to control the image generation process without retraining. However, since these models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders[5]. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity. By introducing cross-attention layers into the model architecture, we turn diffusion models into powerful and flexible generators for general conditioning inputs such as text or bounding boxes and high-resolution synthesis becomes possible in a convolutional manner[2]. Our latent diffusion models (LDMs) achieve a new state of the art for image inpainting and highly competitive performance on various tasks, including unconditional image generation, semantic scene synthesis, and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs[8].

3. EXISTING SYSTEM

Automatic synthesis of realistic images from text would be interesting and useful, but current AI systems are still far from

this goal. However, in recent years generic and powerful recurrent neural network architectures have been developed to learn discriminative text feature representations. Meanwhile, deep convolutional generative adversarial networks (GANs) have begun to generate highly compelling images of specific categories, such as faces, album covers, and room interiors[3]. The development of a novel deep architecture and GAN formulation to effectively bridge these advances in text and image modelling, translating visual concepts from characters to pixels. It demonstrates the capability of our model to generate plausible images of birds and flowers from detailed text descriptions.

Synthesizing high-quality images from text descriptions is a challenging problem in computer vision and has many practical applications. Samples generated by existing text-to-image approaches can roughly reflect the meaning of the given descriptions, but they fail to contain necessary details and vivid object parts[7]. Previously proposed model was Stacked Generative Adversarial Networks (StackGAN) to generate 256x256 photo-realistic images conditioned on text descriptions[3]. The Stage-I GAN sketches the primitive shape and colors of the object based on the given text description, yielding Stage-I low-resolution images. The Stage-II GAN takes Stage-I results and text descriptions as inputs, and generates high-resolution images with photo-realistic details.

4. PROPOSED SYSTEM

Stable Diffusion is a deep learning, text-to-image model. It is primarily used to generate detailed images conditioned on text descriptions, though it can also be applied to other tasks such as inpainting, out painting, and generating image-to-image translations guided by a text prompt[1,2]. Stable Diffusion is a latent diffusion model, and it can run on most consumer hardware equipped with a modest GPU with at least 8 GB VRAM. The text to image sampling script within consumes a text prompt. The script outputs an image file based on the model's interpretation of the prompt. Generated images are tagged with an invisible digital watermark to allow users to identify an image as generated by Stable Diffusion, although this watermark loses its efficacy if the image is resized or rotated[6].

Each generation will involve a specific seed value which affects the output image[2]. Users may opt to randomize the seed in order to explore different generated outputs, or use the same seed to obtain the same image output as a previously generated image. Users are also able to adjust the number of inference steps for the sampler; a higher value takes a longer duration of time; however, a smaller value may result in visual defects. Another configurable option, the classifier-free guidance scale value, allows the user to adjust how closely the output image adheres to the prompt. More experimentative use cases may opt for a lower scale value, while use cases aiming for more specific outputs may use a higher value.

5. METHODOLOGY

The language model creates an embedding of the text prompt. It's fed into the diffusion model together with some random noise. The diffusion model denoises it towards the embedding. This is repeated several times. Then in the end the decoder scales the image up to a larger size[1,2,4]. Figure 1 below shows Latent Diffusion Model.

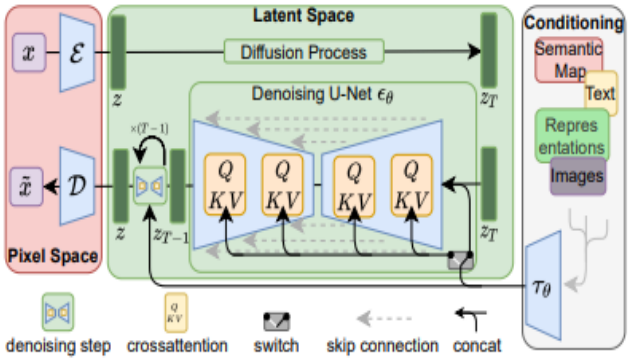

Figure 1: Latent Diffusion Model

5.1 Stages for Image Generation

The image generator goes through two stages:

- **Image information creator**

This component is the secret sauce of Stable Diffusion. Its where a lot of the performance gain over previous models is achieved[2,4].

This component runs for multiple steps to generate image information. This is the *steps* parameter in Stable Diffusion interfaces and libraries which often defaults to 50 or 100.

5

The image information creator works completely in the *image information space* (or *latent* space). We'll talk more about what that means later in the post. This property makes it faster than previous diffusion models that worked in pixel space. In technical terms, this component is made up of a UNet neural network and a scheduling algorithm.

The word "diffusion" describes what happens in this component. It is the step-by-step processing of information that leads to a high-quality image being generated in the end (by the next component, the image decoder).

- **Image Decoder**

The image decoder paints a picture from the information it got from the information creator. It runs only once at the end of the process to produce the final pixel image.

With this we come to see the three main components (each with its own neural network) that make up Stable Diffusion[1,2,4]: Figure 2 below shows Logical flow of stable diffusion.

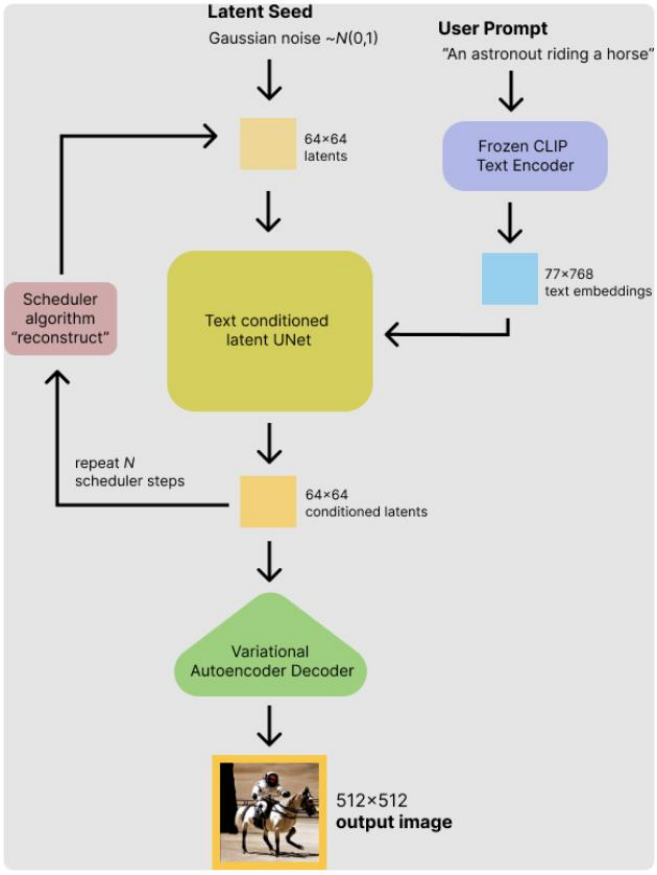

Figure 2: Logical flow of stable diffusion

5.2 The autoencoder (VAE)

The VAE model has two parts, an encoder and a decoder. The encoder is used to convert the image into a low dimensional latent representation, which will serve as the input to the *U-Net* model.

The decoder, conversely, transforms the latent representation back into an image During latent diffusion *training*, the encoder is used to get the latent representations (*latents*) of the images for the forward diffusion process, which applies more and more noise at each step. During *inference*, the denoised latents generated by the reverse diffusion process are converted back into images using the VAE decoder. As we will see during inference, we **only need the VAE decoder[1,2,3]**.

5.3 The U-Net

The U-Net has an encoder part and a decoder part both comprised of ResNet blocks. The encoder compresses an image representation into a lower resolution image representation and the decoder decodes the lower resolution image representation back to the original higher resolution image representation that is supposedly less noisy. More specifically, the U-Net output predicts the noise residual which can be used to compute the predicted denoised image representation[1,2,4].

To prevent the U-Net from losing important information while downsampling, short-cut connections are usually added between the downsampling ResNets of the encoder to the upsampling ResNets of the decoder. Additionally, the stable diffusion U-Net is able to condition its output on text-embeddings via cross-attention layers. The cross-attention layers are added to both the encoder and decoder part of the U-Net usually between ResNet blocks.

5.4 The Text-encoder

The text-encoder is responsible for transforming the input prompt, *e.g.* "An astronaut riding a horse" into an embedding space that can be understood by the 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 and simply uses an CLIP's already trained text encoder, CLIPTextModel[5].

6. IMPLEMENTATION

6.1 Software requirements

- Python 3.9
- Python libraries: PyTorch, Tkinter, CustomTkinter, BERT Tokenizer, Diffusers.
- VS Code

6.2 Hardware requirements

- A modern Intel or AMD CPU.
- 8 GB RAM
- A GPU with the video memory of 6 GB of above

6.3 Tkinter UI

Figure 3 below shows the UI using Tkinter.

Figure 3: UI using Tkinter

7. RESULTS

Figure 4: Results

Figure 4 above the results obtained.

8. CONCLUSION

Latent diffusion models, a simple and efficient way to significantly improve both the training and sampling efficiency of de-noising diffusion models without degrading their quality. Based on this and a cross-attention conditioning mechanism, these experiments could demonstrate favourable results compared to state-of-the-art methods across a wide range of conditional image synthesis tasks without task-specific architectures.

REFERENCES

1. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. and Ommer, B., ***High-Resolution Image Synthesis with Latent Diffusion Models*** ,2022. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
2. Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, Ilya Sutskever, ***Zero-Shot Text-to-Image Generation.***
3. Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee, ***Generative Adversarial Text to Image Synthesis.***
4. Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, ***GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models.***
5. Aditya Ramesh, Prafulla Dhariwa, Casey Chu, ***Hierarchical Text-Conditional Image Generation with CLIP Latents.***
6. Chitwan Saharia, William Chan, Saurabh Saxena, ***Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding.***
7. Prafulla Dhariwal , Alex Nichol, ***Diffusion Models Beat GANs on Image Synthesis.***
8. Danilo Jimenez Rezende, Shakir Mohamed, Daan Wierstra, ***Stochastic Backpropagation and Approximate Inference in Deep Generative Models.***