DOI: https://doi.org/10.36910/6775-2524-0560-2025-61-08
UDC 004.415.3
Pekh Petro, PhD
https://orcid.org/0000-0002-6327-3319
Frolov Oleksandr, Master of Science in Computer Engineering
Lutsk National Technical University, Lutsk, Ukraine

RESEARCH ON THE TECHNOLOGY OF IMAGE GENERATION BASED ON TEXT DESCRIPTION USING THE STABLE DIFFUSION MODEL

Pekh P., Frolov O. Research on the technology of image generation based on text description using the Stable Diffusion model. The article discusses the results of the research on the technology of image generation based on text description using the Stable Diffusion model. The principles of the forward and reverse diffusion mechanism, which consists in gradually adding and removing noise from images, were considered in detail. The implementation was carried out using the Python programming language and the PyTorch and Hugging Face diffusers libraries, which allowed for effective generation of images from given text queries. A software module was developed that demonstrates the operation of the Stable Diffusion architecture. The module implements a full generation cycle - from entering a text query by the user to obtaining a finished image. The components U–Net, Variational Autoencoder (VAE) and CLIP text encoder were used to build the system. The created module allows you to set the generation parameters (number of diffusion steps, level of text influence, resolution, etc.) and visualizes the results obtained. The forward and backward diffusion algorithms underlying the model have been investigated. Based on experiments, it has been found that reducing the number of diffusion steps preserves image quality provided that the noise coefficients and the guidance scale parameter are correctly selected. It has also been confirmed that the use of latent space allows for a significant reduction in computational costs without losing the photorealism of the result.
Keywords: Stable Diffusion, Clip Text Encoder, U-Net & Scheduler, AutoEncoder & Decoder, Tocen

Пех П. А., Фролов О. Д. Дослідження технології генерації зображень за текстовим описом засобами дифузійної моделі Stable Diffusion. У статті розглядаються результати дослідження технології генерації зображень на основі текстового опису із використанням дифузійної моделі Stable Diffusion. Було детально розглянуто принципи роботи механізму прямої та зворотної дифузії, який полягає у поступовому додаванні та видаленні шуму із зображень. Реалізацію здійснено з використанням мови програмування Python та бібліотек PyTorch і Hugging Face diffusers, що дозволило ефективно виконати генерацію зображень із заданих текстових запитів. Розроблено програмний модуль, який демонструє роботу архітектури Stable Diffusion. Модуль реалізує повний цикл генерації – від введення текстового запиту користувачем до отримання готового зображення. Для побудови системи використано компоненти U–Net, Variational Autoencoder (VAE) та текстовий енкодер CLIP. Створений модуль дозволяє задавати параметри генерації (кількість кроків дифузії, рівень впливу тексту, роздільну здатність тощо) і візуалізує отримані результати. Досліджено алгоритми прямої та зворотної дифузії, що лежать в основі роботи моделі. На основі експериментів виявлено, що зменшення кількості кроків дифузії зберігає якість зображення за умови правильного підбору коефіцієнтів шуму та параметра guidance scale. Також підтверджено, що використання латентного простору дозволяє суттєво зменшити обчислювальні витрати без втрати фотореалістичності результату.
Ключові слова: Stable Diffusion, Clip Text Encoder, U-Net & Scheduler, AutoEncoder & Decoder, Tocen Embedding

The problem statement consists in creating and researching a software module that implements the process of image generation based on text description (text–to–image generation) using diffusion models [1 - 6]. To do this, it is necessary to study the technology of diffusion models, the processes of forward diffusion (adding noise) and reverse diffusion (gradual image restoration), which will allow implementing a basic generation model, for example, Stable Diffusion, on a local machine and performing visualization and testing of the results.

There are different classes of image generation models [7 - 11]:

– generative adversarial networks (GAN) – learn through rivalry between the generator and the discriminator, but often have unstable convergence.

– variational autoencoders (VAE) – provide controlled generation, but are inferior in the visual quality of the results.

– diffusion models (DM) – form images by multi-stage noise removal, achieving photorealism and stability of learning.

The main idea of diffusion models is that the generation process is the reverse of the process of gradual noisiness. First, the model learns to add noise to real images, and then - to restore them, reproducing the structure of the original data. Thus, instead of directly generating the image from scratch, the system "denoises" random noise, gradually bringing it closer to the realistic image. The process can be formally described through forward and reverse diffusion:

Forward diffusion. In the forward diffusion stage, the model gradually adds noise to the original image until it receives a completely noisy image, which contains almost no information about the original. This is described by the equation [7,8]:

$$(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \quad (1)$$

The result is a sequence of noisy images, which is used to train the model in the reverse direction.

Back diffusion. During generation, the model performs the reverse process: it starts with random noise and gradually restores the structure of the original image. For this, a neural network is used that learns to predict the noise at each step, and the image restoration is described as [7,8]:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad (2)$$

where x_0 – is the initial (clean) image,
x_T – is the fully noisy image,
β_t– is the diffusion coefficient at step t,
θ– is the model parameters being trained.
For clarity, the process is shown in Figure 1.

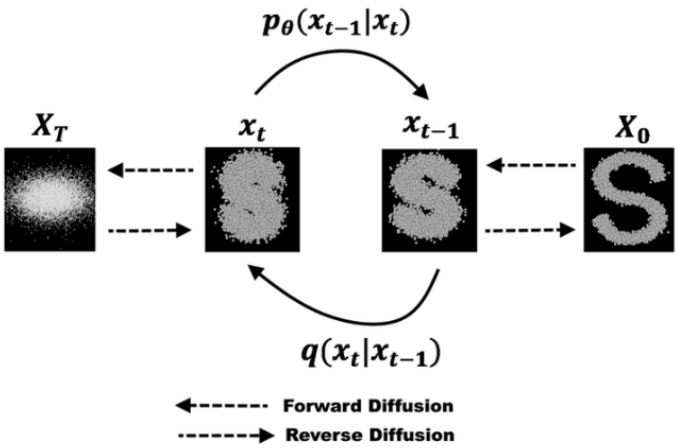

Fig. 1 Scheme of the forward and reverse diffusion process [7]

Thus, to achieve the goal, a diffusion approach was chosen, which combines high stability of learning, controllability of the generation process and the possibility of scaling to large data sets. In the following sections, the methodology of model implementation, development environment and results of experiments on image generation will be considered.

The methodology of implementing diffusion models is based on the gradual transformation of random noise into a structured image using a trained neural network. The main idea is a two-stage process - forward diffusion and back diffusion, which mathematically describe the movement of data in the probability space. [1,2]

During training, the model receives noisy images and tries to predict the noise that has been added.

This encourages the network to accurately reproduce the structure of the data, even from highly noisy input examples. After training, during generation, this process occurs in the opposite direction - the model gradually "cleans" the noise, creating a realistic image from scratch.

Stable Diffusion implements this principle by using a U–Net for image processing, a Variational Autoencoder (VAE) to transform images into latent space, and a text encoder (usually CLIP) to interpret textual cues.

During training, the model is fed noisy images and attempts to predict the noise that has been added.

This forces the network to accurately reproduce the structure of the data, even from very noisy input examples. After training, during generation, this process is reversed - the model gradually "cleans up" the noise, creating a realistic image from scratch. Stable Diffusion implements this principle by using a U–Net for image processing, a Variational Autoencoder (VAE) to transform images into latent space, and a text encoder (usually CLIP) to interpret textual cues.

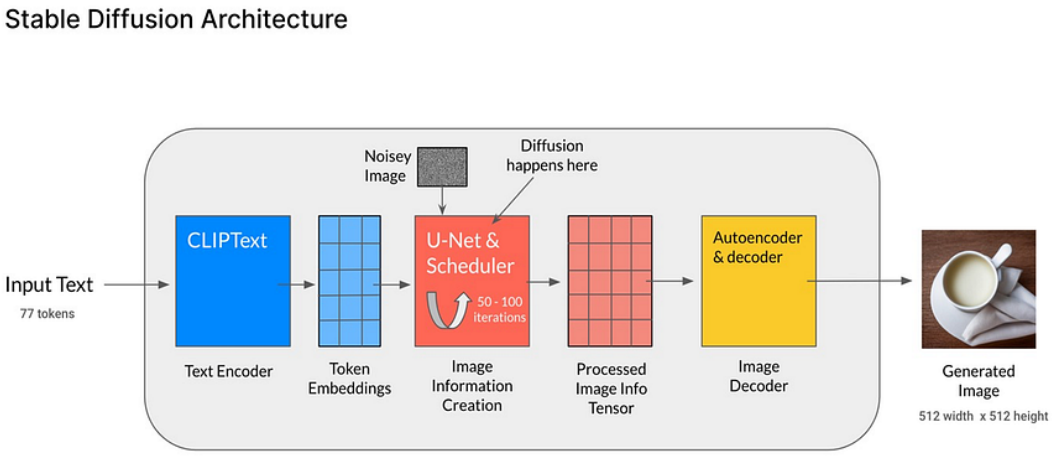

Fig. 2 Stable Diffusion Architecture [8]

Implementation algorithm in practice:

1. The user enters a text query (prompt).
2. The encoder (CLIP) converts the text into a vector representation.
3. Random noise is created in the latent space.
4. U–Net gradually "cleans" the noise, focusing on the text description.
5. The VAE decoder converts the latent representation into the final image.

Tools and development environment. To implement and study diffusion models, in particular the Stable Diffusion architecture, a modern development environment with support for computing on graphics processors (GPU) is used. This allows significantly accelerating the process of image generation and neural network training.

The development was carried out on a local machine with the PyTorch, Transformers and Diffusers libraries installed. The GPU was used to optimize calculations, however, in the absence of CUDA support, execution on a CPU with lower performance is possible.

Main libraries and their purpose:

– PyTorch – used to build, train and execute deep neural networks.

– Hugging Face diffusers – provides a high-level API for working with diffusion models (Stable Diffusion, DDIM, DDPMScheduler, etc.).

– Transformers – contains pre-trained text encoders, such as CLIP, used for prompt processing.

– Accelerate – allows you to effectively scale work on GPU or CPU.

– Matplotlib / PIL – are used to visualize the generated results.

After installing the libraries, the correctness of the configuration is checked. To do this, you can execute the following commands in python (Fig. 3). If we get the result 2.3.1, then the environment is configured correctly and ready for image generation.

```
>>> import torch
>>> print(torch.__version__)
2.7.1+cu118
>>> print("CUDA доступна:", torch.cuda.is_available())
CUDA доступна: True
>>>
```

Fig. 3 PyTorch and CUDA test result in the terminal

Research of the Stable Diffusion model. For the practical implementation of the diffusion model, Stable Diffusion was chosen as one of the most common architectures for generating images based on text descriptions. The model works in latent space, which allows reducing computational costs without losing the quality of the result.

In order to use Stable Diffusion via python, you need to run the code (Fig. 4)

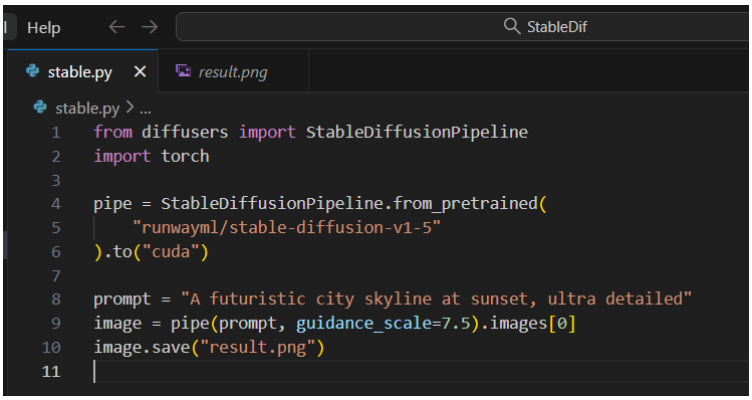

Fig. 4 Code for image generation

During code execution, the model goes through the stages described in Figure 2, after which it generates an image that matches the given description (Figure 5). When re-running, the result may differ slightly due to random initialization of noise.

Fig. 5 Generated image "result.png"

To evaluate the performance of the implemented solution, a series of experiments were conducted using a fixed set of prompts and deterministic seeds. The evaluation focused on the following aspects:

– generation time under different schedulers;
– stability and reproducibility of generated images.

Four schedulers were tested: DDIM, Euler, Heun, and DPM++. For each scheduler, ten images were generated with identical prompts to observe consistency. The number of inference steps was set to 30 for faster testing, and classifier-free guidance strength was fixed.

Tab. I Results of experimental research of generation performance

Scheduler	Avg. Time per Image (s)	Std. Dev.	Notes
DDIM	4.8	0.2	Fastest, slightly reduced detail.
Euler	5.6	0.3	Balanced quality and speed.
Heun	6.1	0.3	Stable with sharp edges
DPM++	6.4	0.4	Best detail, slowest

To complement the numerical evaluation, several qualitative experiments were performed using different diffusion schedulers. All examples were generated with fixed seeds and identical prompt settings to ensure consistent comparison. Representative results are shown in Figures 6–8.

Fig. 6 Example output generated using Euler, 512×512, balanced mode

Figure 6 demonstrates a typical output produced by the Euler scheduler at 512×512 resolution. The result exhibits balanced sharpness and semantic consistency, which aligns with the expected behavior of Euler-based samplers.

A direct comparison between Euler A and DPM++ at a higher resolution of 640×640 is presented in Figure 7. DPM++ produces sharper fine structures and more stable texture reconstruction, while Euler A tends to smooth high-frequency regions. This observation supports the quantitative differences reported earlier.

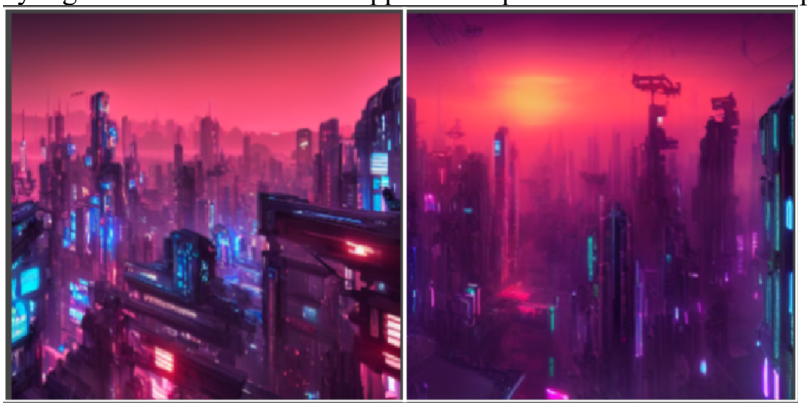

Fig. 7 Comparison of Euler A (right) and DPM++ (left) at 640×640 resolution

Figure 8 illustrates the output generated using the DDIM scheduler under a quality-oriented profile. While DDIM provides the fastest inference, some minor detail smoothing is visible, consistent with its deterministic update rule.

Fig. 8 DDIM output produced under a quality-oriented generation profile

These visual results confirm the trade-offs identified in the performance measurements and highlight the practical differences between commonly used diffusion schedulers. The results show that DDIM provides the fastest generation, while DPM++ yields the highest level of detail, consistent with typical behaviour reported in diffusion literature. The Stable Diffusion model has demonstrated the ability to generate high-quality, detailed images from text descriptions. Thanks to open libraries and GPU support, this implementation can be used as a basis for further research or your own generative experiments.

Conclusions. Diffusion models are a significant step in the development of generative artificial intelligence, as they combine mathematical rigor and practical efficiency. Due to the process of step-by-step noise reduction and data restoration, they provide stable learning and the ability to reconstruct high-quality images. Formal description through forward and backward diffusion equations using equations allows you to accurately model the transformation of random noise into structured visual data. This approach opens the way to creating modern models with the possibility of their wide practical application for image generation tasks.

References

1.Ho J., Jain A., Abbeel P. Denoising Diffusion Probabilistic Models. In: Advances in Neural Information Processing Systems. 34th Conference (NeurIPS 2020), Vancouver, Canada. 2020. [PDF] Available: https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b–Paper.pdf. Proceedings NeurIPS+2arXiv+2

2.Dehouche N. What's in a text–to–image prompt? The potential of Stable Diffusion. Patterns. 2023. Vol. 4, No. 5. DOI/Publisher. [Electronic resource] Available: https://www.sciencedirect.com/science/article/pii/S2405844023039646. ScienceDirect

3.Podell D., English Z., Lacey K. et al. SDXL: Improving Latent Diffusion Models for High–Resolution Image Synthesis. arXiv preprint. 2023. [Electronic resource] Available: https://arxiv.org/abs/2307.01952. arXiv

4.An Introduction to Diffusion Models and Stable Diffusion." Marvik AI Blog. 2023, Nov 28. [Electronic resource] Available: https://blog.marvik.ai/2023/11/28/an–introduction–to–diffusion–models–and–stable–diffusion/. Marvik

5.«Рисунок 1.1 – Схематичне зображення процесу прямої та зворотної дифузії у DDPM». ResearchGate. [Electronic image resource] Available: https://www.researchgate.net/figure/Overview–of–the–forward–and–inverse–diffusion–processes–of–the–DDPM–The–forward_fig3_380875223.

6.«Рисунок 1.2 – Механізм cross–attention у Stable Diffusion». ResearchGate. [Electronic image resource] Available: https://www.researchgate.net/figure/Visualization–of–cross–attention–maps–estimated–by–Stable–Diffusion–The–redder–pixels_fig2_376080892.

7.«Рисунок 1 – Схема процесу прямої та зворотної дифузії». ResearchGate. [Electronic image resource] Available: https://www.researchgate.net/figure/Process–of–Denoising–Diffusion–Probabilistic–Model–Image–by–author_fig1_373932597.

8.«Рисунок 2 – Архітектура Stable Diffusion». Medium. [Electronic image resource] Available: https://medium.com/@alextakele16/image–generation–using–stable–diffusion–mo–b3cf65481692.

9.Nichol A., Dhariwal P. Improved Denoising Diffusion Probabilistic Models. *arXiv preprint* arXiv:2102.09672. 2021. Available: https://arxiv.org/abs/2102.09672

10. Saharia C., Chan W., Ho J. et al. Imagen: Photorealistic Text–to–Image Diffusion Models. *arXiv preprint* arXiv:2205.11487. 2022. Available: https://arxiv.org/abs/2205.11487

11. Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B. High–Resolution Image Synthesis with Latent Diffusion Models. *arXiv preprint* arXiv:2112.10752. 2022. Available: https://arxiv.org/abs/2112.10752