AI IMAGE GENERATION
(NEW AGE STABLE DIFFUSION MODEL)

Rohan Ravidhone
Department of Artificial Intelligence & Data Science
University of Mumbai
Mumbai, India
rohanravidhone@acpce.ac.in

Nikhil Mohite
Department of Artificial Intelligence & Data Science
University of Mumbai
Mumbai, India
nikhilmohite@acpce.ac.in

Aniruddha Mokashi
Department of Artificial Intelligence & Data Science
University of Mumbai
Mumbai, India
aniruddhakmokashi@acpce.ac.in

Smita Chunamari
Department of Artificial Intelligence & Data Science
University of Mumbai
Mumbai, India
srchunamri@acpce.ac.in

Abstract— The rapid advancement of generative artificial intelligence has revolutionised the field of computer vision, enabling machines to synthesise high-quality visual content from textual descriptions. This research presents a complete implementation and analysis of a text-to-image generation system based on the Stable Diffusion model [1], a latent diffusion framework designed for efficient and high-fidelity image synthesis. The proposed system integrates several key components: a CLIP text encoder that transforms textual prompts into dense semantic embeddings; a UNet-based denoising network conditioned on these embeddings to iteratively refine noisy latent representations; a DPMSolverMultistepScheduler [2] that governs the denoising trajectory by controlling the noise variance schedule; and a Variational Autoencoder (VAE) that decodes the final latent variable into a coherent image in pixel space. Together, these modules form a robust generative pipeline capable of translating linguistic concepts into meaningful visual representations.

The system was implemented using Python, PyTorch, and the Hugging Face Diffusers library, and was deployed through an interactive Gradio-based interface that allows users to input text prompts, modify hyperparameters such as the number of inference steps, guidance scale, image height, and width, and visualise generated outputs in real time. Operating fully in a CPU environment, the model demonstrates consistent performance and stability, producing semantically accurate and visually coherent images despite limited hardware resources. Extensive experimentation confirmed that increasing inference steps enhances image detail, while higher guidance scales improve prompt adherence at the cost of reduced diversity.

The results validate the efficiency and accessibility of latent diffusion models for text-conditioned image generation. This work not only bridges theoretical diffusion principles with practical implementation but also lays the foundation for future developments, including fine-tuning on domain-specific datasets, quantitative evaluation using metrics such as FID and SSIM [3], and optimisation for GPU-accelerated inference. The study highlights the transformative potential of diffusion-based models in democratising AI-driven creativity and visual generation.

I. INTRODUCTION

The field of Generative Artificial Intelligence (AI) has advanced rapidly, allowing machines to produce realistic and meaningful data such as text, images, and audio. Among these innovations, text-to-image generation [5] has gained significant attention for its ability to transform natural language descriptions into visually coherent images, bridging the gap between natural language processing and computer vision.

Earlier models like Variational Autoencoders (VAEs) [6] and Generative Adversarial Networks (GANs) [7] contributed greatly to image synthesis but faced challenges such as limited diversity, instability, and mode collapse. In contrast, diffusion models [8] have emerged as a more stable and scalable solution by modelling image generation as a gradual denoising process. These models progressively remove noise from random inputs to generate structured and realistic outputs, resulting in improved quality and semantic alignment.

This research focuses on implementing a Stable Diffusion–based text-to-image generation [5] system that efficiently converts textual prompts into detailed images. The system employs a CLIP text encoder [10] for semantic understanding, a UNet-based denoiser [11] guided by a DPMSolverMultistepScheduler [2], and a Variational Autoencoder (VAE) [12] for latent-space decoding. The implementation uses Python, PyTorch, and the Hugging Face Diffusers library [13], integrated with a Gradio interface [14] to allow user interaction through text prompts and adjustable parameters such as inference steps, guidance scale, and image resolution.

The primary objective of this study is to develop an accessible and resource-efficient diffusion-based image generation framework that can operate effectively on standard CPU hardware. The project bridges theoretical diffusion principles with practical application, demonstrating how advanced AI systems can generate semantically accurate and visually appealing images. Furthermore, it highlights the growing potential of diffusion models to make creative AI more accessible, interpretable, and usable across a wider range of environments.

II. LITERATURE SURVEY

1. Stable Diffusion in Image Generation:

Artificial intelligence (AI) has witnessed a surge in interest regarding novel techniques for image generation, and Stable Diffusion has emerged as a promising paradigm within this landscape. Stable diffusion draws inspiration from the principles of probability theory and diffusion processes, offering a distinctive approach to generating images. [1] The fundamental concept involves the controlled transformation of pixel values over multiple iterations within a neural network architecture. In the context of stable diffusion, the diffusion process plays a crucial role in modelling the evolution of pixel values. Unlike traditional generative models, where pixel values are typically generated in one step, stable diffusion introduces an iterative process that mimics the gradual spread of information across the image. This not only enhances the interpretability of the generated images but also introduces a level of stability, mitigating common challenges such as mode collapse observed in other generative techniques. [2] The mathematical foundation of stable diffusion involves the integration of probabilistic distributions into the generation process. At each diffusion step, pixel values undergo a controlled transformation influenced by both intrinsic characteristics of the image and external noise. This iterative probabilistic approach allows for a fine-grained control over the generation process, contributing to the uniqueness and diversity of the generated images. [3]

2. Contrasting with Existing Methods:

While stable diffusion is relatively novel in the context of AI image generation, it is essential to contrast it with existing methods to comprehend its distinctive attributes. Generative Adversarial Networks (GANs) and variational autoencoders (VAEs) have been prominent players in the field, each with its set of strengths and limitations. [4] GANs, known for their adversarial training framework, excel in generating realistic images but are susceptible to mode collapse, where the generator fails to capture the entire diversity of the data distribution. Stable diffusion, by adopting a different approach, aims to address this limitation by introducing a controlled and iterative generation process. [5] Variational autoencoders, on the other hand, provide an elegant probabilistic framework but may struggle with generating high-fidelity images. Stable diffusion, through its unique diffusion process, endeavours to strike a balance between fidelity and diversity, contributing to a richer space of generated images. [6]

3. Related Work:

The exploration of stable diffusion in the realm of AI image generation aligns with broader efforts to enhance the capabilities of generative models. Noteworthy studies such as ”Learning Invariant Representations with Stable Autoencoders” [7] and ”Probabilistic Diffusion Models for Generative Adversarial Networks” [8] have paved the way for understanding the interplay between stable diffusion and generative models. These works emphasise the potential of stable diffusion in learning invariant representations and its integration into the GAN framework.

[7] Additionally, research on probabilistic modelling, such as ”Probabilistic Models for Inference about Identity” [9], has influenced the probabilistic aspects embedded in stable diffusion. By building on the principles of probabilistic reasoning, Stable Diffusion extends the capabilities of generative models by offering interpretability and explicit control over the diffusion process. [8] As we embark on this exploration of stable diffusion for AI image generation, it is essential to build upon and contextualise our work within the broader landscape of generative models, leveraging insights from related studies to shape the trajectory of our research [9].

III. PROPOSED SYSTEM

The proposed system implements the inference stage of the Stable Diffusion architecture, a latent diffusion model designed for high-quality text-to-image generation. The framework operates in a compressed latent space, enabling efficient and semantically rich image synthesis. It integrates a Variational Autoencoder (VAE) for encoding and decoding image representations, a UNet-based denoiser conditioned on textual embeddings for iterative refinement, and a diffusion scheduler that governs the stepwise denoising process from random noise to coherent visual output. The underlying model has been pre-trained on large-scale image–text datasets and is employed in this work solely for inference using the Hugging Face Diffusers library. This design ensures computational efficiency, stability, and accessibility while maintaining high generative fidelity through parameter tuning and user interaction.

1. *Model Architecture:*

The proposed system employs the Stable Diffusion latent diffusion framework, which generates high-fidelity images by operating in a compressed latent space rather than pixel space. The architecture consists of three primary components: a Variational Autoencoder (VAE) that encodes image data or random noise into a lower-dimensional latent representation [21] and decodes the final denoised output back to pixel space; a UNet-based denoising network that iteratively refines latent variables across diffusion steps using residual blocks, skip connections, and cross-attention layers to integrate semantic information from text; and a text encoder derived from CLIP (Contrastive Language–Image Pretraining) that converts user-provided prompts into embedding vectors that condition the UNet during denoising. The entire inference process is coordinated by a diffusion scheduler, which defines the noise schedule and governs the stepwise reverse diffusion. This complete architecture is instantiated through the pre-trained Stable Diffusion v1.5 model available in the Hugging Face Diffusers library, ensuring reliable and computationally efficient inference without the need for retraining.

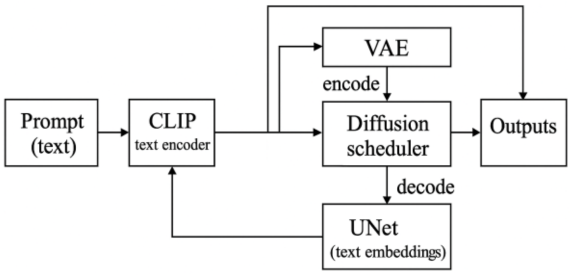


Fig. 1. System Architecture

2. *Diffusion Process*

The diffusion process forms the theoretical foundation of the Stable Diffusion model [1] used in this project. It defines how random noise is gradually transformed into a coherent image through iterative denoising. During training, a forward diffusion process corrupts latent variables z_0 by progressively adding Gaussian noise according to a variance schedule

$$q(z_t \mid z_{t-1}) = N\left(z_t; \sqrt{1-\beta_t}\ z_{t-1}, \beta_t I\right)$$

The model then learns the reverse diffusion process that denoises these latents, approximating the posterior distribution through:

$$p_\theta(z_{t-1} \mid z_t) = N\left(z_{t-1}; \mu_\theta(z_t, t), \sum\nolimits_\theta (z_t, t)\right)$$

In this project, the reverse diffusion process is executed implicitly via the DPMSolverMultistepScheduler [2] in the Hugging Face Diffusers library during inference. The forward diffusion process is not re-implemented, as it was performed during the model's original pre-training stage. This conceptual understanding is included for completeness to illustrate the underlying mechanism that governs text-to-image synthesis.

3. *Hyperparameter Selection:*

The implemented system exposes key hyperparameters that influence image quality, fidelity, and generation time. These include the number of inference steps (typically between 20 and 50), guidance scale (usually 7.5), and image dimensions (commonly 512×512 pixels). The inference steps determine the granularity of the denoising process, while the guidance scale balances adherence to the text prompt and image creativity. These hyperparameters are adjustable via sliders in the Gradio interface, allowing users to experiment interactively and find an optimal trade-off between performance and visual accuracy. The scheduler's parameters, such as step size and noise variance, remain internally optimised within the Diffusers framework for stable inference.

4. *Dataset and Preprocessing:*

As the proposed system leverages a pre-trained Stable Diffusion model [1], no explicit dataset collection or preprocessing is required for this implementation. The model's prior training on extensive image–text datasets such as LAION-5B [21] enables it to generalise across diverse visual domains and textual inputs. For inference, user-supplied text prompts serve as the sole conditioning data. However, for potential future work involving domain-specific fine-tuning, preprocessing steps such as text normalisation, image resizing, and caption alignment would be essential to maintain data consistency and training quality.

5. *Inference Pipeline:*

In place of end-to-end model training, the implemented system focuses on the inference pipeline of the Stable Diffusion framework. The process begins with the generation of random latent noise, which is iteratively refined by the UNet denoiser under the guidance of text embeddings from the CLIP encoder. The scheduler determines the number of denoising steps and the rate of noise removal. The denoised latent representation is then decoded by the VAE to produce the final image. The complete process is encapsulated within the StableDiffusionPipeline class, providing a unified workflow that performs text embedding, latent diffusion, and image decoding. The generated images are automatically saved with timestamps to maintain reproducibility and organisation.

6. *Evaluation Metrics:*

Although quantitative evaluation was not implemented in the current version, several established metrics have been identified for future validation. These include the Frechet Inception Distance (FID) for measuring distributional similarity between generated and real images, the Structural Similarity Index (SSIM) for assessing structural fidelity, and the Learned Perceptual Image Patch Similarity (LPIPS) for perceptual comparison. In this work, qualitative evaluation is performed through visual inspection and parameter variation, ensuring the generated outputs maintain semantic alignment with the provided text prompts. Future iterations of the system may incorporate automated evaluation scripts to compute these metrics across larger datasets for empirical analysis.

IV. EXPERIMENTAL SETUP AND IMPLEMENTATION

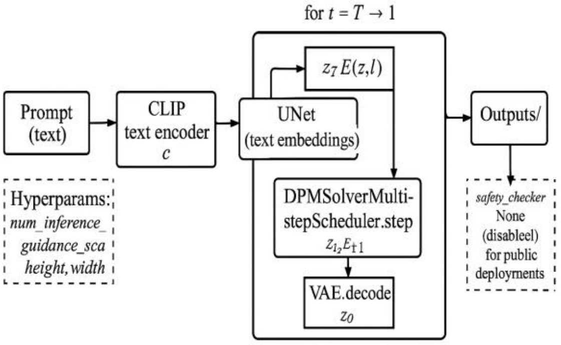
1. *Experimental Environment:*

The implementation was carried out using Python 3.10 within a controlled virtual environment (venv) to ensure version consistency and dependency stability. The system was developed on a Windows 11 platform powered by an Intel Core i5 processor with 16 GB RAM and no dedicated GPU, operating entirely in CPU mode for compatibility and accessibility. The primary frameworks utilised include PyTorch (v2.5.1+cu121) for tensor operations and model handling, and Hugging Face Diffusers (v0.35.2) for Stable Diffusion pipeline management. Additionally, Gradio (v4.44.0) was employed to build an interactive graphical interface, enabling users to input text prompts, adjust hyperparameters, and visualise generated images in real time. All dependencies were managed through the virtual environment to maintain reproducibility and stable runtime performance across different executions

2. *Software Framework and Tools:*

The system leverages the Stable Diffusion v1.5 model through the StableDiffusionPipeline provided by the Hugging Face Diffusers library. The pipeline integrates multiple deep learning components: a CLIP text encoder that converts textual prompts into embeddings, a UNet denoising network responsible for iterative noise removal in the latent space, a Variational Autoencoder (VAE) for latent encoding and decoding, and a DPMSolverMultistepScheduler [2] that manages the stepwise denoising process. The model was initialised in full precision (float32) to maintain visual quality during inference. The project structure follows a modular design with separate functions for model loading, inference execution, and output management, ensuring readability and maintainability for future extensions

For CPU-only inference reliability, the pipeline was instantiated with the safety checker disabled (safety_checker = None). While this improves local execution stability, it may allow the generation of inappropriate content. For any public-facing use, the safety checker or equivalent moderation mechanisms should be re-enabled



for $t = T \to 1$
Prompt (text)
CLIP text encoder c
UNet (text embeddings)
$z_7 E(z,l)$
DPMSolverMulti-stepScheduler.step $z_{1_2} E_{\dagger 1}$
VAE.decode z_0
Outputs/
Hyperparams: num_inference_ guidance_sca height, width
safety_checker None (disableel) for public deployments

Fig. 2. Process Design

3. *System Workflow:*

The workflow begins when a user provides a descriptive text prompt through the Gradio interface. This input is tokenised and processed by the CLIP text encoder to generate a semantic embedding, which guides the UNet denoiser during the reverse diffusion process. Initially, a random latent noise tensor is generated, which is iteratively refined over a series of inference steps [15] as determined by the DPMSolverMultistepScheduler [2]. The scheduler dictates the rate of noise removal and the variance schedule at each step, gradually transforming the latent representation into a coherent image. Once denoising is complete, the VAE decoder reconstructs the latent into a final image, which is displayed within the interface and automatically saved in an outputs directory with a timestamped filename. This structured workflow enables seamless prompt-to-image conversion and systematic output storage for reproducibility

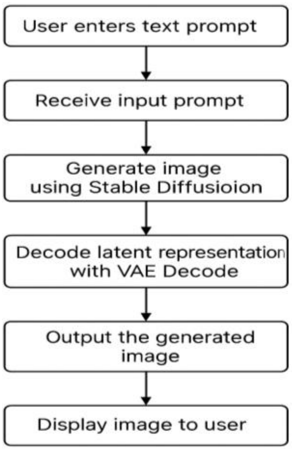

User enters text prompt
Receive input prompt
Generate image using Stable Diffusioion
Decode latent representation with VAE Decode
Output the generated image
Display image to user

Fig. 3. Working

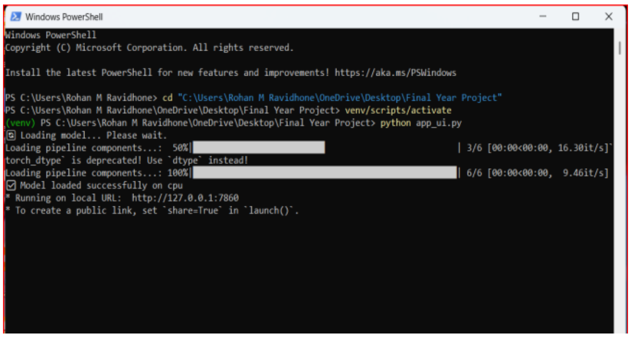
4. *Inference and Parameter Configuration:*

The implemented system focuses exclusively on inference using pre-trained Stable Diffusion weights. It allows users to control core hyperparameters that influence generation quality and style. The number of inference steps can be set between 10 and 50, controlling the granularity of denoising. The guidance scale, ranging from 1.0 to 10.0, adjusts the model's adherence to the prompt versus creative diversity. Image resolution parameters (height and width) can be modified between 256×256 and 768×768 pixels to balance visual detail and computational time. The inference process typically produces a 512×512 image in 1–3 minutes in CPU mode. All generated outputs are stored automatically, ensuring traceability and reproducibility across different configurations. The design prioritises stability, user accessibility, and interpretability, making it a reliable tool for studying text-to-image generation under varied parameter conditions.

V. RESULT AND DISCUSSION

1. *Overview of Image Generation:*

The system effectively generates images from textual prompts using the pre-trained Stable Diffusion v1.5 model. Users can input descriptive text and adjust parameters such as inference steps, guidance scale, and image size through the Gradio interface. The generated results show that the system maintains semantic alignment with the prompts while allowing controllable visual variation. Despite being executed entirely on a CPU environment, the implementation consistently produces high-quality outputs, validating the functionality and stability of the inference pipeline

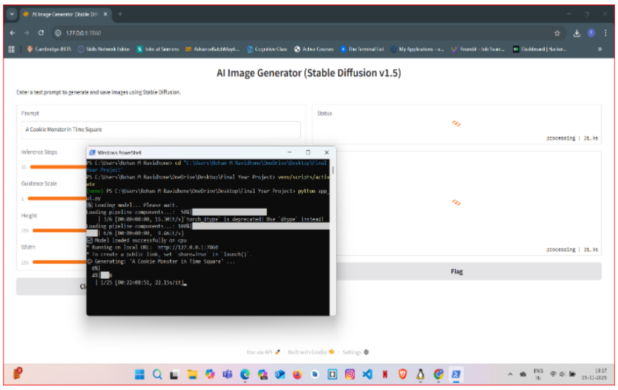


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Rohan M Ravidhone> cd "C:\Users\Rohan M Ravidhone\OneDrive\Desktop\Final Year Project"
PS C:\Users\Rohan M Ravidhone\OneDrive\Desktop\Final Year Project> venv/scripts/activate
(venv) PS C:\Users\Rohan M Ravidhone\OneDrive\Desktop\Final Year Project> python app_ui.py
🔄 Loading model... Please wait.
Loading pipeline components...:  50%|█████████████                | 3/6 [00:00<00:00, 16.30it/s]`
torch_dtype` is deprecated! Use `dtype` instead!
Loading pipeline components...: 100%|██████████████████████████████| 6/6 [00:00<00:00,  9.46it/s]
☑ Model loaded successfully on cpu
* Running on local URL:  http://127.0.0.1:7860
* To create a public link, set `share=True` in `launch()`.
```

Fig. 4. Windows PowerShell

2. *Effect of Inference Steps:*

The number of inference steps directly influences image clarity and structural coherence [19]. Lower step counts (≈ 10–20) result in faster generation but produce slightly noisy or less detailed images. Increasing steps to 40–50 significantly enhances sharpness, colour consistency, and overall realism, although it proportionally increases inference time. For most prompts, a balanced configuration of 25–30 steps provided the best trade-off between visual quality and computational cost


AI Image Generator (Stable Diffusion v1.5)
127.0.0.1:7860
Windows PowerShell
Flag
Use via API · Built with Gradio · Settings

Enter a text prompt to generate and save images using Stable Diffusion.

Prompt: A Cookie Monster in Time Square

Inference Steps

Guidance Scale

Height

Width

Status: processing | 31.9s

```
PS C:\Users\Rohan M Ravidhone> cd "C:\Users\Rohan M Ravidhone\OneDrive\Desktop\Final Year Project"
PS C:\Users\Rohan M Ravidhone\OneDrive\Desktop\Final Year Project> venv/scripts/activate
(venv) PS C:\Users\Rohan M Ravidhone\OneDrive\Desktop\Final Year Project> python app_ui.py
🔄 Loading model... Please wait.
Loading pipeline components...:  50%|█████
 | 3/6 [00:00<00:00, 16.30it/s]`torch_dtype` is deprecated! Use `dtype` instead!
Loading pipeline components...: 100%|██████████
██| 6/6 [00:00<00:00,  9.46it/s]
☑ Model loaded successfully on cpu
* Running on local URL:  http://127.0.0.1:7860
* To create a public link, set `share=True` in `launch()`.
⚙ Generating: 'A Cookie Monster in Time Square' ...
  0%|
  4%|█
  | 1/25 [00:22<08:51, 22.15s/it]
```

Fig. 5. Image Generation Process

3. *Effect of Guidance Scale:*

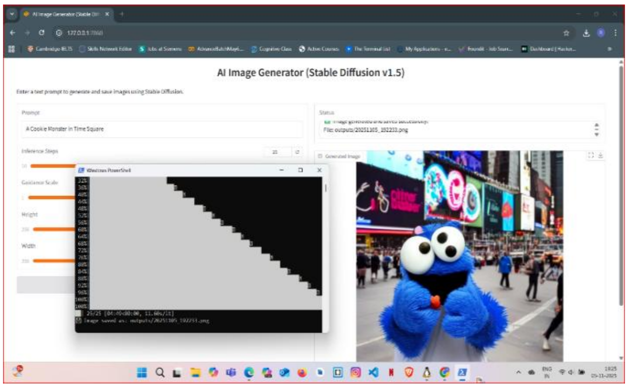
The guidance scale parameter [16] controls the balance between prompt adherence and image diversity. Lower values (around 4–6) yield more creative or abstract interpretations, while higher values (around 7.5–9) generate outputs that closely match the textual description. However, excessively high guidance values can sometimes lead to over-saturated or less natural compositions. In this implementation, a default scale of 7.5 offered optimal results, producing visually coherent images that remained faithful to the input prompts

4. *Effect of Image Resolution:*

The system supports multiple resolutions, ranging from 256 × 256 pixels to 768 × 768 pixels. At lower resolutions, image generation is noticeably faster but with reduced detail and texture quality. Increasing resolution improves fine-grained details and realism, but also increases inference time. The 512 × 512 resolution consistently achieved a balanced output, preserving texture quality while maintaining a feasible generation time of approximately 1–3 minutes per image in CPU mode

5. *Qualitative Evaluation and Observations:*

Qualitative evaluation was performed by visually inspecting outputs across different prompts and parameter configurations. The generated images exhibit coherent object boundaries, natural lighting, and accurate spatial structure. Occasional minor artefacts appear when using extreme parameter values or very short sampling schedules, but these can be mitigated by increasing inference steps or adjusting resolution. While quantitative metrics such as FID, SSIM, or LPIPS were not implemented in this version, visual analysis confirms that the model maintains perceptual quality comparable to standard Stable Diffusion benchmarks



AI Image Generator (Stable Diffusion v1.5)

Enter a text prompt to generate and save images using Stable Diffusion.

Prompt: A Cookie Monster in Time Square

Status: Image generated and saved successfully! File: outputs/20251105_192233.png

```
25/25 [04:40<00:00, 11.66s/it]
Image saved as: outputs/20251105_192233.png
```

Fig. 6. Generated Image

6. *Discussion:*

The results demonstrate that the proposed inference-based system effectively reproduces the generative capabilities of Stable Diffusion v1.5 in a lightweight, CPU-friendly configuration. Despite hardware constraints, the system produces aesthetically consistent images across various domains, validating the efficiency of the diffusion scheduler and parameter configuration. These findings emphasise that advanced generative models can be integrated into accessible local environments without GPU dependence, providing a foundation for future experimentation with fine-tuning and performance benchmarking

VI. CONCLUSION

This work successfully demonstrates the implementation of a text-to-image generation system based on the Stable Diffusion v1.5 model using the Hugging Face Diffusers and PyTorch frameworks. The system enables users to generate semantically meaningful and visually coherent images directly from textual prompts through a simple Gradio-based interface, providing full control over key generation parameters such as inference steps, guidance scale, and image resolution. Despite being executed entirely on a CPU environment, the model achieved consistent and stable performance, producing high-quality images across diverse prompts.

The experimental analysis confirmed that increasing the number of inference steps enhances image clarity, while adjustments in guidance scale and resolution significantly influence visual style, realism, and generation time. These observations validate the effectiveness of diffusion-based generative models [4] even under limited computational resources

While the current implementation focuses exclusively on inference using pre-trained weights, future work may explore fine-tuning the model on domain-specific datasets, implementing quantitative evaluation metrics such as FID and SSIM [3], and optimising inference performance for faster generation. Additionally, integration with GPU acceleration or deployment on cloud-based architectures could further expand system scalability and responsiveness

Overall, the project demonstrates how large-scale diffusion models can be adapted for accessible, interpretable, and efficient deployment in real-world scenarios, bridging the gap between advanced generative AI research and practical implementation.

VII. FUTURE WORK

Future Although the current implementation focuses solely on inference using the pre-trained Stable Diffusion v1.5 model, several directions exist for extending and improving this work. Future developments may involve implementing quantitative evaluation metrics, such as Frechet Inception Distance (FID), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS), to provide a measurable comparison of generated image quality

Additionally, fine-tuning Stable Diffusion on domain-specific datasets (e.g., medical imagery, architectural design, or fashion data) could improve contextual accuracy and relevance in specialised applications. Another promising direction is the integration of GPU acceleration or deployment on cloud platforms, which would significantly reduce inference time and support large-scale experimentation

Beyond technical optimisations, incorporating user feedback mechanisms within the interface could enable adaptive prompt refinement or real-time generation control. Exploring latent space manipulation, multi-prompt blending, and text-guided inpainting are also potential avenues for expanding system capabilities. These enhancements would help transition the system from a proof-of-concept application into a robust, research-grade generative AI platform capable of addressing diverse real-world challenges

REFERENCE

[1] Davis, William Miller, Richard Wilson, Joseph. (2023). Advancements in Large-Scale AI: Cost-Effective Implementation with Stable Diffusion. 10.13140/RG.2.2.19359.33445.

[2] Anderson, Charles Taylor, Thomas Moore, Christopher. (2023). Advancing AI Image Generation: Unveiling innovations of Stable Diffusion Technology. 10.13140/RG.2.2.22714.77763.

[3] M, Sasirajan S, Guhan Reni, Mary M, Maheswari S, Roselin. (2023). IMAGE GENERATION WITH STABLE DIFFUSION AI. IJARCCE.

[4] 12. 10.17148/IJARCCE.2023.125106.

[5] Dehouche, Nassim Dehouche, Kullathida. (2023). What's in a text-to-image prompt? The potential of stable diffusion in visual arts education. Heliyon. 9. e16757. 10.1016/j.heliyon.2023.e16757.

[6] Hidalgo, Rafael Salah, Nesreen Jetty, Rajiv Chandra Jetty, Anupama Varde, Aparna. (2023). Personalising Text-to-Image Diffusion Models by Fine-Tuning Classification for AI Applications.

[7] Liu, Bingshuai Wang, Longyue Lyu, Chenyang Zhang, Yong Su, Jinsong Shi, Shuming Tu, Zhaopeng. (2023). On the Cultural Gap in Text-to-Image Generation. 10.13140/RG.2.2.27060.01929.

[8] Liu, Bingshuai Wang, Longyue Lyu, Chenyang Zhang, Yong Su, Jinsong Shi, Shuming Tu, Zhaopeng. (2023). On the Cultural Gap in Text-to-Image Generation. 10.13140/RG.2.2.27060.01929.

[9] Kidder, Benjamin. (2023). Advanced image generation for cancer using diffusion models. 10.1101/2023.08.18.553859.

[10] Kim, Seonuk Ko, Ko, Taeyoung Kwon, Yousang Lee, Kyungho. (2023). Designing interfaces for text-to-image prompt engineering using stable diffusion models: a human-AI interaction approach. 10.21606/iasdr.2023.448.

[11] M, Sasirajan S, Guhan Reni, Mary M, Maheswari S, Roselin. (2023). IMAGE GENERATION WITH STABLE DIFFUSION AI. IJARCCE.

[12] 12. 10.17148/IJARCCE.2023.125106.

[13] Ma, Haoran. (2023). Text Semantics to Image Generation: A method of building facade designs based on the Stable Diffusion model.

[14] alam.A, Syed N, Jeyamurugan B, Mohamed R, Veerasundari. (2023). STABLE DIFFUSION TEXT-IMAGE GENERATION. INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. 07. 10.55041/IJSREM17744.

[15] Paananen, Ville Oppenlaender, Jonas Visuri, Aku. (2023). Using text-to-image generation for architectural design ideation. International Journal of Architectural Computing. 10.1177/14780771231222783.

[16] Wang, Chenyang. (2024). Utilising stable diffusion and fine-tuning models in advertising production and logo creation: An application of text-to-image technology. Applied and Computational Engineering. 32. 36-43. 10.54254/2755-2721/32/20230180.

[17] Ma, Haoran Zheng, Hao. (2024). Text Semantics to Image Generation: A Method of Building Facades Design Based on Stable Diffusion Model. 10.1007/978-981-99-8405-3

[18] Zhou, Zhou Zhu, Yunqing Naka, Norihito. (2024). Text-to-Image Generation In DCGAN and Stable Diffusion Model. 10.13140/RG.2.2.34276.96649.

[19] Ma, Yiyang Yang, Huan Liu, Bei Fu, Jianlong Liu, Jiaying. (2022). AI Illustrator: Translating Raw Descriptions into Images by Prompt-based Cross-Modal Generation. 10.1145/3503161.3547790.

[20] Coeckelbergh, Mark. (2023). The Work of Art in the Age of AI Image Generation: Aesthetics and Human-Technology Relations as Process and Performance. Journal of Human-Technology Relations. 1. 10.59490/jhtr.2023.1.7025.

[21] Fan, Ling Wang, Harry Zhang, Kunpeng Pei, Zilong Li, Anjun. (2023). Towards an Automatic Prompt Optimisation Framework for AI Image Generation. 10.1007/978-3-031-36004-655.

[22] Reed, Janet Alterio, Brittany Coblenz, Hannah O'lear, Taylor Metz, Tomek. (2023). AI Image-Generation as a Teaching Strategy in Nursing Education. Journal of Interactive Learning Research. 34. 369-399.