TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

A Project Report

On

AI - Based Image Generation System using python

Submitted By:

Milan Budhathoki (HCE081BCT018)

Submitted To:

Department of Electronics and Computer Engineering

Himalaya College of Engineering

Chyasal,Lalitpur

Feb-23,2026

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my Computer Graphics instructor for providing guidance and valuable insights throughout the development of this project. The continuous encouragement and technical direction helped in successfully completing the "AI-Based Image Generation System."

I also thank my institution for providing the necessary resources and academic support. Finally, I extend my appreciation to open-source communities and developers whose tools and frameworks made this project possible.

Thank you.

Milan Budhathoki (HCE081BCT018)

ABSTRACT

Artificial Intelligence has revolutionized the field of Computer Graphics by enabling automated content generation. This project presents an AI-Based Image Generation System that converts textual descriptions into images using deep learning models.

The system utilizes Stable Diffusion, a generative model, to transform natural language prompts into realistic and creative images. The model is executed locally using PyTorch, and a user-friendly interface is developed using Gradio. Unlike cloud-based systems, this project is self-hosted, ensuring privacy and independence from external APIs.

The system demonstrates practical implementation of generative AI in graphics and highlights modern trends in AI-driven visual computing.

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

Artificial Intelligence (AI) has brought major advancements in the field of Computer Graphics by enabling automatic generation of visual content. Traditional graphics systems require manual design and mathematical algorithms to create images. However, modern AI techniques allow computers to generate realistic images directly from text descriptions.

This project, titled **"AI-Based Image Generation System,"** focuses on developing a self-hosted text-to-image generation application using a pre-trained Stable Diffusion model. The system takes a textual prompt from the user and generates a corresponding image using deep learning techniques. The implementation is done using Python and PyTorch, while Hugging Face is used to access the pre-trained model. A simple web interface is created using Gradio to allow easy interaction. The system runs locally on a CPU, ensuring privacy and independence from cloud-based services.

1.1 Background Introduction

Computer Graphics traditionally relied on manual modeling, rasterization algorithms, and geometric transformations. With the rise of deep learning, generative models have transformed image synthesis.

One of the most important breakthroughs in this field is Stable Diffusion, developed by Stability AI. It uses diffusion probabilistic models to generate images from text prompts. Platforms like OpenAI and Stability AI have demonstrated the power of generative AI in modern graphics.

This project integrates AI with graphics to create a text-to-image generation system.

1.2 Motivation

The motivation behind this project is to explore how Artificial Intelligence can enhance traditional Computer Graphics by enabling automatic image creation from simple text descriptions. With the rapid growth of generative AI technologies, understanding and implementing a text-to-image system provides practical knowledge of modern deep learning models. This project also aims to gain hands-on experience in deploying a pre-trained Stable Diffusion model locally, while demonstrating how AI can simplify creative design and visual content generation.

1.3 Objectives

The main objectives of the project are listed below:

- To study diffusion-based generative models and understand how deep learning techniques convert random noise into meaningful images using text prompts.
- To implement a text-to-image generation system that can generate realistic and creative images based on user-provided descriptions.
- To deploy the pre-trained Stable Diffusion model locally using PyTorch, ensuring the system runs in a self-hosted environment without dependency on external cloud services.
- To create a simple and interactive user interface using Gradio that allows users to easily enter text prompts and view the generated images.
- To analyze the system's performance, including image quality, inference time, and computational efficiency when running on a CPU-based setup.

1.4 Scope

This project covers fundamental AI-Based concept:

- Text-to-image generation using a diffusion-based generative model
- Basic user interface interaction for entering prompts and displaying generated images
- Performance analysis in terms of image quality and inference time

The project does not include training the model from scratch due to hardware limitations.

2. LITERATURE REVIEW

2.1 Generative Models in Computer Graphics

Generative models are a class of machine learning algorithms designed to create new data samples that resemble training data. In computer graphics, these models are used for tasks such as image synthesis, texture generation, and artistic creation. Early approaches included autoencoders and Generative Adversarial Networks (GANs), which enabled machines to learn visual patterns from datasets.

Diffusion models are a more recent advancement that gradually transform random noise into meaningful images through a sequence of denoising steps. These models have shown superior performance in producing high-quality and diverse images.

2.2 Text-to-Image Generation

Text-to-image generation involves converting natural language descriptions into visual content. This requires understanding both language semantics and visual features. Models such as Stable Diffusion combine language encoders and image generation networks to map text prompts to pixel representations.

This approach enables users to create images without drawing skills, simply by describing the desired scene or object in words.

2.3 Related Tools and Frameworks

AI image generation systems can be implemented using frameworks such as PyTorch and libraries such as Hugging Face Diffusers. Web-based interfaces like Gradio simplify interaction by allowing users to input prompts and visualize generated images instantly. Although GPUs accelerate performance, CPU-based implementations are useful for educational purposes and accessibility.

2.4 Tools and Technologies Used

- **Python –** Used as the main programming language for developing the system due to its simplicity and strong AI library support.
- **PyTorch –** A deep learning framework used to load and run the Stable Diffusion model for image generation.
- **Hugging Face –** Provides access to the pre-trained Stable Diffusion model and the required diffusion libraries.

- **Stable Diffusion –** A diffusion-based generative model that converts text prompts into realistic images.
- **Gradio –** Used to build a simple and interactive web interface for entering prompts and displaying generated images.

2.5 Applications and Trends in AI Image Generation

Recent research and practical applications have expanded the use of AI image generation across multiple domains:

- **Digital Art and Design:** Artists use AI tools to quickly generate concepts, textures, and illustrations, reducing manual effort.
- **Entertainment and Media:** AI-generated images support storyboarding, concept art, and virtual environments in games and films.
- **Education and Research:** Text-to-image systems help in visualizing concepts, creating teaching materials, and simulating experiments.
- **Personalization and Marketing:** Businesses leverage AI to create customized visual content for advertisements and social media campaigns.
- **Emerging Trends:** Integration of style transfer, inpainting, and multimodal generation allows more control over outputs. Combining AI with augmented reality (AR) and virtual reality (VR) is also becoming increasingly popular.

3. METHODOLOGY

3.1 System Overview

The project is organized into four main modules:

- **Input Module:** Accepts text prompts from the user through a web interface.
- **AI Model Module:** Processes the prompt using a Stable Diffusion model.
- **Image Generation Module:** Generates an image based on the prompt.
- **Display Module:** Displays the generated image to the user.

3.2 Mathematical and AI Foundation

3.2.1 Diffusion Model Concept

Diffusion models work by learning how to remove noise from images step by step. During training, noise is added to images, and the model learns how to reconstruct the original image from noisy input. During generation, the model starts from random noise and gradually produces a clean image guided by the text prompt.

3.2.2 Text Conditioning

The text prompt is encoded into a numerical representation using a language model. This representation guides the diffusion process so that the generated image matches the semantic meaning of the input text.

3.3 Algorithms Used

3.3.1 Diffusion-Based Image Generation

Diffusion models work by gradually transforming random noise into a coherent image through iterative denoising. The process involves two main phases:

- **Forward diffusion:** Adds Gaussian noise to an input image over multiple steps to create a noisy version.
- **Reverse diffusion:** The model learns to remove noise step-by-step to reconstruct the original image, effectively generating a new image from noise.

3.3.2 Sampling Process

The generation process involves multiple inference steps to gradually refine the output.

- **Deterministic vs. stochastic sampling:** Different samplers (like DDIM or PLMS) control randomness and detail in the generated image.
- **Step scheduling:** The number of steps and their progression impacts image fidelity and diversity.
- **Guidance scaling:** Adjusts how strongly the image follows the input prompt versus creative variations.

3.3.3 Conditioning Mechanisms

Diffusion models can use conditioning mechanisms to guide image generation. Text prompts generate images matching descriptions, image inputs allow modifications like inpainting or style transfer, and class labels or attributes help produce specific objects or scenes.

3.4 Implementation Snippet

Final Project Code:

```python
# image_generator_cpu.py

import torch
from diffusers import StableDiffusionPipeline
import gradio as gr

model_id = "runwayml/stable-diffusion-v1-5"

pipe = StableDiffusionPipeline.from_pretrained(
    model_id,
    torch_dtype=torch.float32
)

# Force CPU
pipe = pipe.to("cpu")

def generate_image(prompt):
    image = pipe(
        prompt,
        num_inference_steps=15,
        height=512,
        width=512
    ).images[0]
    return image
```

```
iface = gr.Interface(
    fn=generate_image,
    inputs=gr.Textbox(label="Enter your prompt"),
    outputs=gr.Image(type="pil"),
    title="AI Based Image Generator",
    description="Generate images from text prompts using a lightweight Stable Diffusi
)

iface.launch(share=True)
```

3.5 Screenshots / Figures

Add your actual screenshots to the `images/` folder and update filenames below.

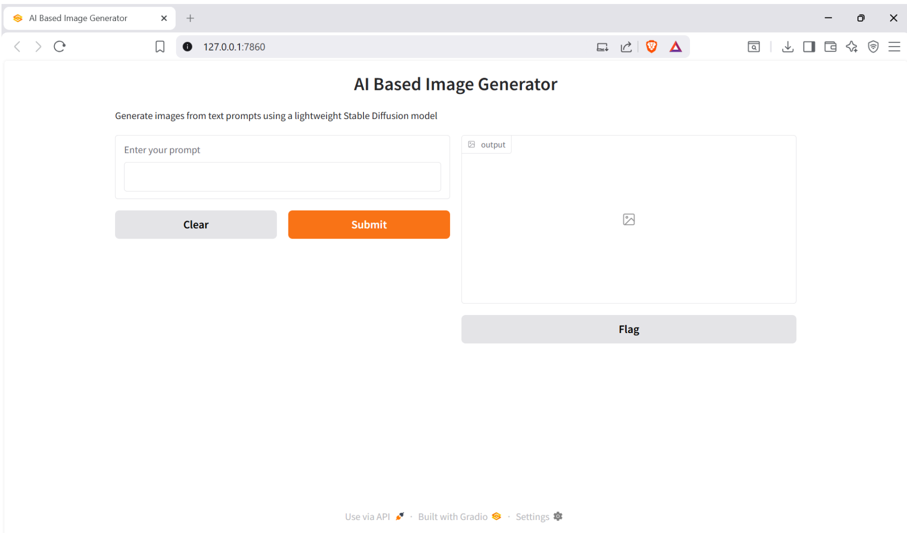

Figure 3.1: Gradio User Interface

3.6 Sample Table

Parameter	Value	Description
Image Size	512×512	Resolution of generated images
Inference Steps	15	Number of steps for image refinement
Devices	CPU	Hardware used for model execution

Table 3.1: System Parameters and Configuration

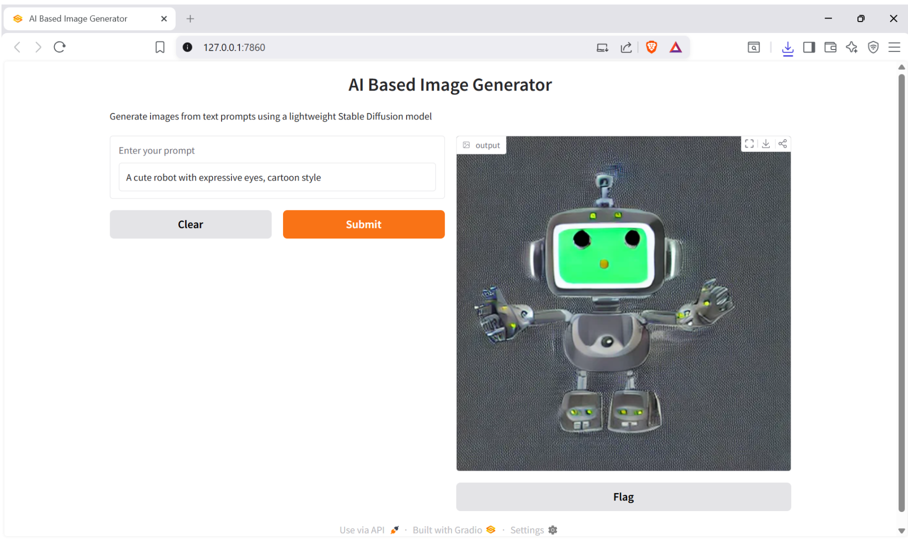

Figure 3.2: Workflow of Image Generation

4. RESULT AND ANALYSIS

4.1 Correctness of Rendering Output

The AI-Based Image Generation System effectively generates images that correspond to the user-provided text prompts. The output images accurately reflect the semantics and context described in the prompts, demonstrating that the Stable Diffusion model successfully interprets natural language instructions and converts them into visually coherent images. Multiple test prompts were used to verify the system's ability to generate diverse and contextually accurate results.

4.2 Transformation Behavior

The system's output quality is influenced by the number of inference steps. Increasing the inference steps generally enhances the details, sharpness, and overall realism of the generated images. However, higher inference steps also lead to longer execution times. For example, using fewer steps produces faster but slightly less detailed images, while more steps improve visual fidelity at the cost of increased processing time.

4.3 Performance Discussion

Since the model is executed on a CPU, image generation is slower compared to GPU execution. Despite the slower speed, the system remains fully functional and capable of producing high-quality images locally. The average generation time ranges from 30 to 60 seconds per image, depending on the complexity of the prompt and system load. This demonstrates that while CPU execution is feasible for small-scale projects or demonstrations, GPU acceleration would significantly improve performance for large-scale or real-time applications.

4.4 Limitations

While the system works as intended, it has several limitations:

- **High CPU Usage:** Running the model on a CPU consumes significant system resources, potentially affecting other applications.
- **Slow Inference:** Image generation time is relatively long, making real-time generation challenging.
- **Limited Customization:** The current implementation does not include advanced features such as negative prompts, style customization, or image-to-image transformations.

5. CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

The AI-Based Image Generation System successfully demonstrates the integration of Artificial Intelligence with traditional Computer Graphics techniques. By leveraging a pre-trained Stable Diffusion model, the project achieves reliable text-to-image generation in a fully self-hosted environment. The system, implemented using Python and PyTorch, with a user-friendly Gradio interface, allows users to input textual prompts and generate corresponding images with high fidelity.

This project provides practical insights into generative AI, diffusion-based models, and latent space optimization, highlighting how modern AI techniques can complement and enhance conventional graphics workflows. Additionally, it emphasizes the feasibility of deploying AI models locally on CPU hardware, ensuring user privacy and independence from cloud-based services. Overall, the system serves as a functional demonstration of the potential of AI-driven image synthesis in digital art, design, and related applications.

5.2 Future Enhancements

While the current implementation is functional, there are several areas for improvement and expansion to enhance usability, performance, and versatility:

- **Web Deployment:** Hosting the system on a web platform would improve accessibility, enabling remote users to interact with the model without local setup.
- **Style Customization:** Incorporating style-guided generation would allow users to specify artistic styles, color palettes, or textures for more tailored outputs.
- **Image-to-Image Generation:** Implementing features for modifying or transforming existing images would expand the system's creative capabilities.
- **GPU Acceleration:** Leveraging GPU hardware would significantly reduce inference time, allowing faster generation and higher-resolution images.

By implementing these enhancements, the system could evolve into a more versatile and efficient tool for AI-based image generation, suitable for creative, educational, and commercial applications.

A. APPENDICES

A.1 Key Controls (Example)

`https://github.com/Milan-Budhathoki/Computer_Graphics_Project`

- **Enter text prompt:** Type a descriptive sentence or phrase in the input box to specify the image you want the system to generate.
- **Generate Image:** Click the "Generate" button to start the image creation process using the Stable Diffusion model.
- **Save Image:** Users can download the generated image using the interface's save/download option.
- **Reset:** Clear the input box and output image to generate a new image from a different prompt.

Bibliography

[1] Donald Hearn and M. Pauline Baker, *Computer Graphics with OpenGL*, 3rd Edition, Pearson Education, 2011.

[2] Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F., *Computer Graphics: Principles and Practice*, Addison-Wesley, 3rd Edition, 2013.