**Stable Diffusion Image Generation**

**Project Overview**

This project uses **Stable Diffusion**, a powerful generative AI model, to create high-quality images from text prompts. The model is implemented using **Hugging Face's diffusers library** and deployed with **Gradio** on **Hugging Face Spaces**.

**Features**

* Text-to-image generation using **Stable Diffusion v1.5**.
* Optimized for **CPU & GPU** execution.
* Simple **web-based UI** using **Gradio**.
* **Free deployment** on Hugging Face Spaces.

**Tools & Frameworks Used**

| **Category** | **Tool/Framework Used** | **Why We Used It?** |
| --- | --- | --- |
| **Model** | diffusers (Hugging Face) | Provides **pre-trained** Stable Diffusion models with **easy-to-use API**. |
| **Deep Learning** | torch (PyTorch) | Efficient GPU/CPU execution and **widely used** for AI/ML research. |
| **Model Optimization** | accelerate (Hugging Face) | Improves inference speed, supports **multi-GPU acceleration**. |
| **Text Processing** | transformers (Hugging Face) | Handles **text embeddings** for prompt-based generation. |
| **Web UI** | Gradio | Simple UI to deploy and interact with models **without frontend coding**. |
| **Deployment** | Hugging Face Spaces | Free cloud hosting for AI models with **easy integration**. |

**Approach Followed**

| **Step** | **Approach Used** | **Why This Approach?** |
| --- | --- | --- |
| **Model Selection** | Used **Stable Diffusion v1.5** from diffusers | Open-source, **pre-trained**, and **efficient** for image generation. |
| **Pipeline Setup** | Used StableDiffusionPipeline from diffusers | Provides **a clean API** to load and generate images. |
| **Device**  **Optimization** | Checked torch.cuda.is\_available() | Uses GPU if available, **falls back to CPU** otherwise. |
| **UI Development** | Used Gradio for interface | **No frontend coding required**, quick to deploy. |
| **Deployment** | Hugging Face Spaces | Free hosting, **no setup hassle**, easy to share. |

**Why This Approach vs. Other Alternatives?**

| **Category** | **Our Choice** | **Alternative** | **Why We Chose Our Approach?** |
| --- | --- | --- | --- |
| **Model** | diffusers | stable-diffusion-webui | diffusers is **lightweight, scriptable**, and easier to integrate with Python. |
| **Deep Learning Framework** | torch (PyTorch) | tensorflow | PyTorch is **better for research and flexibility** in Generative AI. |
| **UI Framework** | Gradio | Streamlit | Gradio is **simpler**, better for **AI/ML models** with interactive UI. |
| **Deployment** | Hugging Face Spaces | Google Colab, AWS | Spaces provides **free GPU options**, is **easier** to maintain than AWS. |

**Installation & Setup**

**1️. Clone the Repository**

git clone https://github.com/OutOfLearning/Stable-Diffusion-Image-Generator.git

cd stable-diffusion-image-generation

**2️. Create & Activate Virtual Environment**

# Windows

type nul > requirements.txt

python -m venv venv

venv\Scripts\activate

# Linux/Mac

python3 -m venv venv

source venv/bin/activate

**3️. Install Dependencies**

pip install -r requirements.txt

**4️. Run the Application**

python app.py

**How It Works**

1. Enter a **text prompt** in the Gradio UI.
2. Click **"Generate"** to create an image.
3. The AI model processes the prompt and generates an image.
4. Download or save the generated image.

**Deployment on Hugging Face Spaces**

To deploy the app on Hugging Face Spaces:

gradio deploy

Follow the instructions and choose the **CPU/GPU option** for best performance.

**FAQ**

**Did we use an API key?**

No, we used a **public pre-trained model** from Hugging Face, so no API key was required.

**Why is image generation slow?**

* **On CPU** → It takes more time (use GPU for better performance).
* **On free Hugging Face Spaces** → Limited resources (upgrade to GPU if needed).

**Can I customize the model?**

Yes! You can fine-tune the Stable Diffusion model with your own dataset.

**License**

This project is open-source under the **MIT License**.

**Author**

Developed by **Vamsi Krishna Prasad Manchiraju**

**Acknowledgments**

* Hugging Face for diffusers and transformers
* PyTorch for deep learning
* Gradio for the interactive UI