AI ART with Stable Diffusion in Google Colab

```
# Install necessary libraries
!pip install torch torchvision torchaudio
!pip install diffusers transformers
!pip install gradio
```

```
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.11/di
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/pytho
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/pytho
Requirement already satisfied: gradio in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: aiofiles<25.0,>=22.0 in /usr/local/lib/pyt
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.
Requirement already satisfied: brotli>=1.1.0 in /usr/local/lib/python3.11
Requirement already satisfied: fastapi<1.0,>=0.115.2 in /usr/local/lib/py
Requirement already satisfied: ffmpy in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: gradio-client==1.11.0 in /usr/local/lib/py
Requirement already satisfied: groovy~=0.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: httpx<1.0,>=0.24.1 in /usr/local/lib/pytho
Requirement already satisfied: huggingface-hub<1.0,>=0.33.5 in /usr/local
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/pyt
Requirement already satisfied: numpy<3.0,>=1.0 in /usr/local/lib/python3.
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dis
Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3
Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/pyth
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: python-multipart>=0.0.18 in /usr/local/lib
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3
Requirement already satisfied: ruff>=0.9.3 in /usr/local/lib/python3.11/d
Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in /usr/local/lib/
Requirement already satisfied: semantic-version~=2.0 in /usr/local/lib/py
Requirement already satisfied: starlette<1.0,>=0.40.0 in /usr/local/lib/p
Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in /usr/local/lib/
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/p
Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/p
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dis
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dis
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/pyt
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/p
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.1
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/p
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/py
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/pytho
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/py
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/pytho
```

```
import torch
from diffusers import StableDiffusionPipeline
from PIL import Image
import gradio as gr
```

```
#Create an Access Token on Hugging Face, then store it securely in Google C

from google.colab import userdata
userdata.get('HF_token’)
```

```
from huggingface_hub import login

# Retrieve the Hugging Face API token
hf_token = userdata.get('HF_token')

# Authenticate with Hugging Face
login(token=hf_token)
```

```
# Load the Stable Diffusion model
model_id = "CompVis/stable-diffusion-v1-4"
pipe = StableDiffusionPipeline.from_pretrained(model_id)
pipe = pipe.to("cuda")
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access
  warnings.warn(
```

```
model_index.json: 100%                                   541/541 [00:00<00:00, 53.0kB/s]
Fetching 16 files: 100%                                  16/16 [03:09<00:00, 12.82s/it]
text_encoder/model.safetensors: 100%                     492M/492M [02:26<00:00, 2.63MB/s]
safety_checker/model.safetensors: 100%                   1.22G/1.22G [01:07<00:00, 39.4MB/s]
scheduler_config.json: 100%                              313/313 [00:00<00:00, 10.0kB/s]
config.json: 100%                                        592/592 [00:00<00:00, 13.8kB/s]
preprocessor_config.json: 100%                           342/342 [00:00<00:00, 9.51kB/s]
merges.txt:    525k/? [00:00<00:00, 5.07MB/s]
config.json:    4.56k/? [00:00<00:00, 62.4kB/s]
scheduler_config-                                        209/209 [00:00<00:00, 3.69kB/s]
checkpoint.json: 100%
special_tokens_map.json: 100%                            472/472 [00:00<00:00, 57.6kB/s]
unet/diffusion_pytorch_model.safetensors: 100%           3.44G/3.44G [03:08<00:00, 133MB/s]
vocab.json:    1.06M/? [00:00<00:00, 1.01MB/s]
tokenizer_config.json: 100%                              806/806 [00:00<00:00, 51.5kB/s]
config.json: 100%                                        743/743 [00:00<00:00, 36.8kB/s]
config.json: 100%                                        551/551 [00:00<00:00, 26.0kB/s]
vae/diffusion_pytorch_model.safetensors: 100%            335M/335M [02:30<00:00, 3.00MB/s]
Loading pipeline components...: 100%                     7/7 [00:02<00:00,  2.46it/s]
```

Define the Art Generation Function with Name Overlay

Run the following cell:

```
def generate_image(prompt, num_inference_steps=50, guidance_scale=7.5):
    # Generate the image
    image = pipe(prompt, num_inference_steps=num_inference_steps, guidance
    return image
```

Create a Gradio Interface for Easy Interaction

The Gradio interface. Run the following cell:

```
# Create a Gradio interface
iface = gr.Interface(
    fn=generate_image,
    inputs=gr.Textbox(lines=2, placeholder="Enter your prompt here..."),
    outputs=gr.Image(type="pil"),
    title="Stable Diffusion Art Generator",
    description="Generate art using Stable Diffusion model."
)

# Launch the interface
iface.launch()
```

```
It looks like you are running Gradio on a hosted Jupyter notebook, which re

Colab notebook detected. To show errors in colab notebook, set debug=True i
* Running on public URL: https://ece36e38268eba7346.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgra
```

gradio

No interface is running right now

Example Prompts for Abstract and Gentle Nature

You can use the following prompts to generate images that combine abstract elements with gentle nature scenes:

```
"A serene landscape with soft, abstract shapes and gentle gradients, evoking

"Abstract watercolor patterns blending seamlessly with a peaceful forest scen

"Gentle, flowing lines and abstract forms intertwined with a calming meadow,

"A tranquil lake reflecting abstract, colorful patterns in the sky, with gent

"Abstract, organic shapes and gentle gradients forming a harmonious landscape
```

Running the Art Generator

```
Run the Cells in Order
    First, run the cell to install the necessary libraries.
    Then, run the cell to import the necessary libraries.
    Next, run the cell to authenticate with Hugging Face.
    Then, run the cell to load the Stable Diffusion model.
    Finally, run the cell to define the art generation function and create th

Interact with the Gradio Interface
    Once the Gradio interface is launched, you can enter one of the example p
```

By following these steps, you should be able to run the Stable Diffusion model in Google Colab and generate images using the Gradio interface. If you encounter any issues or need further assistance, feel free to ask!

```
    "Soft-colored circles and gentle gradients in a serene landscape, floatin
```