```
# Install dependencies (run this once per environment)
!pip install diffusers transformers accelerate torch safetensors
!pip install Pillow

# Optional: Log in to Hugging Face if the model requires authentication
from huggingface_hub import login

# Uncomment and enter your Hugging Face token here if needed:
# login("YOUR_HUGGINGFACE_TOKEN")
```

```
Requirement already satisfied: diffusers in /usr/local/lib/python3.12/dist-packages (0.35.2)
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.57.1)
Requirement already satisfied: accelerate in /usr/local/lib/python3.12/dist-packages (1.11.0)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.8.0+cu126)
Requirement already satisfied: safetensors in /usr/local/lib/python3.12/dist-packages (0.6.2)
Requirement already satisfied: importlib_metadata in /usr/local/lib/python3.12/dist-packages (from diffusers) (8.7.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from diffusers) (3.20.0)
Requirement already satisfied: huggingface-hub>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from diffusers) (0.35.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from diffusers) (2.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from diffusers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from diffusers) (2.32.4)
Requirement already satisfied: Pillow in /usr/local/lib/python3.12/dist-packages (from diffusers) (11.3.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.3)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.22.
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: psutil in /usr/local/lib/python3.12/dist-packages (from accelerate) (5.9.5)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from torch) (3.5)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch) (3.1.6)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.7
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch) (9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch) (11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch) (10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch) (11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch) (12.5.4.2)
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in /usr/local/lib/python3.12/dist-packages (from torch) (2.27.3)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.85)
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch) (1.11.1.6)
Requirement already satisfied: triton==3.4.0 in /usr/local/lib/python3.12/dist-packages (from torch) (3.4.0)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub>=0.34.0->di
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch) (1.3.0)
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.12/dist-packages (from importlib_metadata->diffusers) (3.23.
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->torch) (3.0.3)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->diffusers) (3
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->diffusers) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->diffusers) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->diffusers) (2025.10
Requirement already satisfied: Pillow in /usr/local/lib/python3.12/dist-packages (11.3.0)
```

```
# Import required libraries
from diffusers import StableDiffusionPipeline
import torch
from PIL import Image
from IPython.display import display

# Load the Stable Diffusion model
model_id = "runwayml/stable-diffusion-v1-5"

# Use float32 for CPU (float16 only works on GPU)
pipe = StableDiffusionPipeline.from_pretrained(
    model_id,
    torch_dtype=torch.float32,
)
pipe = pipe.to("cpu")  # run on CPU

# Generate an image
prompt = "a cozy cabin in the woods during snowfall"
image = pipe(prompt).images[0]

# Display the image
display(image)
```

```
Loading pipeline components...: 100%                                    7/7 [00:02<00:00,  2.10it/s]

100%                                              50/50 [27:04<00:00, 30.64s/it]
```

```python
from IPython.display import display
import ipywidgets as widgets

prompt_box = widgets.Text(
    value='A magical forest with glowing mushrooms',
    placeholder='Type your image description...',
    description='Prompt:',
    disabled=False
)

button = widgets.Button(description="Generate Image")

output = widgets.Output()

def on_button_clicked(b):
    with output:
        output.clear_output()
        print(f"Generating image for: '{prompt_box.value}'...")
        image = pipe(prompt_box.value).images[0]
        display(image)

button.on_click(on_button_clicked)
display(prompt_box, button, output)
```

Prompt: A magical forest with glowing mus

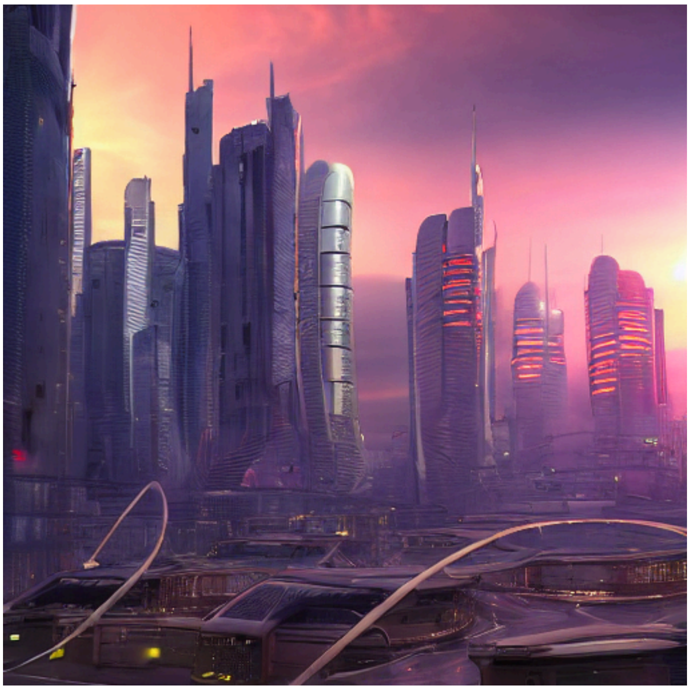
Generate Image

```python
#!/usr/bin/env python3
"""
Quick diagnostic for Stable Diffusion (diffusers) problems.

- Prints Python / package versions and device info.
- Checks for a Hugging Face token in env.
- Attempts to load the pipeline and prints detailed errors and hints.

Usage:
  python debug_stable_diffusion.py
"""
import os
import sys
import traceback
import platform
import torch
import importlib

def print_header(title):
    print("\n" + "="*10 + " " + title + " " + "="*10)

def safe_import(name):
    try:
        m = importlib.import_module(name)
        print(f"{name}: OK ({getattr(m, '__version__', 'unknown')})")
    except Exception as e:
        print(f"{name}: FAILED -> {e}")

def main():
    print_header("System info")
    print("Python:", sys.version.replace("\n"," "))
    print("Platform:", platform.platform())
    print("Torch version:", getattr(torch, "__version__", "not installed"))
    print("CUDA available:", torch.cuda.is_available())
    try:
        print("Default device:", torch.device("cuda" if torch.cuda.is_available() else "cpu"))
    except Exception:
        pass

    print_header("Important packages")
    safe_import("diffusers")
    safe_import("transformers")
    safe_import("accelerate")
    safe_import("safetensors")
    safe_import("PIL")

    print_header("Environment variables (HUGGINGFACE_TOKEN/HF_TOKEN/HF_HOME)")
    for k in ("HUGGINGFACE_HUB_TOKEN", "HUGGINGFACE_TOKEN", "HF_TOKEN", "HF_HOME"):
        print(f"{k} = {os.environ.get(k)}")

    model_id = "runwayml/stable-diffusion-v1-5"
    print_header(f"Try loading pipeline {model_id}")

    try:
        # Import here to capture failures separately
        from diffusers import StableDiffusionPipeline
        torch_dtype = torch.float16 if torch.cuda.is_available() else torch.float32
        kwargs = {}
        # If you have a Hugging Face token in env, use it automatically
        token = os.environ.get("HUGGINGFACE_HUB_TOKEN") or os.environ.get("HF_TOKEN") or os.environ.get("HUGGINGFACE_TOKEN")
        if token:
            kwargs["use_auth_token"] = token

        print("Attempting from_pretrained(...). This may download >4GB. Please be patient.")
        pipe = StableDiffusionPipeline.from_pretrained(model_id, torch_dtype=torch_dtype, **kwargs)
        print("Loaded pipeline object successfully.")
        # move to device
        device = "cuda" if torch.cuda.is_available() else "cpu"
        pipe = pipe.to(device)
        # Enable attention slicing to reduce CPU memory usage
        if hasattr(pipe, "enable_attention_slicing"):
            pipe.enable_attention_slicing()
            print("Enabled attention slicing.")
        else:
            print("Pipeline has no enable_attention_slicing() method (ok).")
        print("Pipeline ready on device:", device)
        # Quick sample run with tiny settings to confirm generation path
        print("Running a tiny sample generation (1 step) to test forward pass...")
        img = pipe("test", num_inference_steps=1, guidance_scale=1.0).images[0]
        print("Sample run OK, image type:", type(img))
    except Exception as e:
        print("ERROR while loading or running pipeline:")
        traceback.print_exc()
        print("\nCommon causes & quick fixes:")
        print("- No internet or download blocked: ensure the machine has network access and huggingface.co is reachable.")
        print("- Missing HF token / model access: set HUGGINGFACE_HUB_TOKEN env var with a valid token or run `huggingface-cli lo
        print("- Not enough memory on CPU: try smaller image sizes, fewer steps, or use a GPU.")
        print("- Incompatible package versions: install stable versions (see below).")
        print("\nSuggested install (CPU):")
        print("  pip install --upgrade pip")
        print("  pip install 'diffusers[torch]' transformers accelerate safetensors pillow")
        print("For torch, follow official instructions at https://pytorch.org/get-started/locally/ (choose CPU or CUDA as appropr
        print("\nIf you still have errors, paste the full traceback above into the chat.")

if __name__ == "__main__":
    main()
```

```
========== System info ==========
Python: 3.12.12 (main, Oct 10 2025, 08:52:57) [GCC 11.4.0]
Platform: Linux-6.6.105+-x86_64-with-glibc2.35
Torch version: 2.8.0+cu126
CUDA available: False
Default device: cpu

========== Important packages ==========
diffusers: OK (0.35.2)
transformers: OK (4.57.1)
accelerate: OK (1.11.0)
safetensors: OK (0.6.2)
PIL: OK (11.3.0)

========== Environment variables (HUGGINGFACE_TOKEN/HF_TOKEN/HF_HOME) ==========
HUGGINGFACE_HUB_TOKEN = None
HUGGINGFACE_TOKEN = None
HF_TOKEN = None
HF_HOME = None

========== Try loading pipeline runwayml/stable-diffusion-v1-5 ==========
Attempting from_pretrained(...). This may download >4GB. Please be patient.
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
model_index.json: 100%                                    541/541 [00:00<00:00, 57.5kB/s]

Fetching 15 files: 100%                                    15/15 [02:44<00:00, 20.65s/it]

special_tokens_map.json: 100%                              472/472 [00:00<00:00, 7.74kB/s]

config.json: 100%                                          617/617 [00:00<00:00, 11.6kB/s]

merges.txt:     525k/? [00:00<00:00, 4.40MB/s]

safety_checker/model.safetensors: 100%                     1.22G/1.22G [02:41<00:00, 5.78MB/s]

preprocessor_config.json: 100%                             342/342 [00:00<00:00, 6.74kB/s]

scheduler_config.json: 100%                                308/308 [00:00<00:00, 5.55kB/s]

text_encoder/model.safetensors: 100%                       492M/492M [02:43<00:00, 4.37MB/s]

config.json:     4.72k/? [00:00<00:00, 75.1kB/s]

tokenizer_config.json: 100%                                806/806 [00:00<00:00, 28.7kB/s]

vocab.json:     1.06M/? [00:00<00:00, 6.42MB/s]

unet/diffusion_pytorch_model.safetensors: 100%             3.44G/3.44G [02:43<00:00, 30.7MB/s]

config.json: 100%                                          743/743 [00:00<00:00, 8.91kB/s]

config.json: 100%                                          547/547 [00:00<00:00, 5.51kB/s]

vae/diffusion_pytorch_model.safetensors: 100%              335M/335M [01:22<00:00, 4.02MB/s]

Loading pipeline components...: 100%                       7/7 [00:02<00:00,  2.86it/s]
`torch_dtype` is deprecated! Use `dtype` instead!
Loaded pipeline object successfully.
```

```
# Import required libraries
from diffusers import StableDiffusionPipeline
import torch
from PIL import Image
from IPython.display import display

# Load the Stable Diffusion model
model_id = "runwayml/stable-diffusion-v1-5"

# Use float32 for CPU (float16 only works on GPU)
pipe = StableDiffusionPipeline.from_pretrained(
    model_id,
    torch_dtype=torch.float32,
)
pipe = pipe.to("cpu")  # run on CPU

# Generate an image with a new prompt
prompt = "a futuristic city at sunset"
image = pipe(prompt).images[0]

# Display the image
display(image)
```

```
Loading pipeline components...: 100%                                  7/7 [00:02<00:00,  2.56it/s]

100%                                     50/50 [21:06<00:00, 24.28s/it]
```



Start coding or generate with AI.