

```
import torch
from diffusers import StableDiffusionPipeline
import os
from PIL import Image
import matplotlib.pyplot as plt
```

Flax classes are deprecated and will be removed in Diffusers v1.0.0. We recommend migrating to PyTorch classes or pinning your version of Diffusers.

Flax classes are deprecated and will be removed in Diffusers v1.0.0. We recommend migrating to PyTorch classes or pinning your version of Diffusers.

Load pre-trained model

```
model_id = "runwayml/stable-diffusion-v1-5"
pipe = StableDiffusionPipeline.from_pretrained(model_id)
pipe = pipe.to("cuda" if torch.cuda.is_available() else "cpu")
```

```
/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 as secret in your Google Colab and restart your session. 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_id": "5e55a06175a848b395ee38401c0f6b05", "version_major": 2, "version_minor": 0}
```

```
{"model_id": "89891703b04843118308b3fddced3272", "version_major": 2, "version_minor": 0}
```

```
{"model_id": "af9fb872d0394e3a9bc5fa9f25878d37", "version_major": 2, "version_minor": 0}
```

```
{"model_id": "83a724193bb94e4e82d7af5146164252", "version_major": 2, "version_minor": 0}
```

```
{"model_id": "815561b639ce4d5998d4da5ab519318d", "version_major": 2, "version_minor": 0}
```

```
{"model_id": "b7c183d2184247328448bae274f417c1", "version_major": 2, "version_minor": 0}
```

```
{"model_id": "23b0b3d874264a25bc7feb3cb38607ab", "version_major": 2, "version_minor": 0}
```

```

{"model_id":"dca06007ffa34d9d9e15eb2efcc6d5a0","version_major":2,"version_minor":0}

{"model_id":"61cf070673a64a80839379211da14e0c","version_major":2,"version_minor":0}

{"model_id":"a3c4d8713ddd465da34a93f5eedfa49a","version_major":2,"version_minor":0}

{"model_id":"7b38dfe517474ca6b304aa4fb12a9480","version_major":2,"version_minor":0}

{"model_id":"f3bc569cab4480294f13626d0841407","version_major":2,"version_minor":0}

{"model_id":"f04ecdb625d14d2aa38d11b85d818bbd","version_major":2,"version_minor":0}

{"model_id":"163597efc54142f99f116b4c1ab5d5ae","version_major":2,"version_minor":0}

{"model_id":"40decef1ff174beaa3923525f8f3732a","version_major":2,"version_minor":0}

{"model_id":"63eac2dabe8f4b53b985eed2b3060f08","version_major":2,"version_minor":0}

{"model_id":"ed0a0b8f3e634b47956a107abb694145","version_major":2,"version_minor":0}

# Create dataset directory
output_dir = "synthetic_image_dataset"
os.makedirs(output_dir, exist_ok=True)

# Input prompts (UPDATED)
prompts = [
    "A cyberpunk Indian smart city with neon lights at night",
    "A humanoid AI robot teaching students using holograms in a classroom",
    "A serene Himalayan landscape during golden hour",
    "A futuristic AI-powered hospital with advanced medical technology",
    "A self-driving electric car on a busy Indian highway during rain"
]

# Generate images
for i, prompt in enumerate(prompts):
    image = pipe(prompt).images[0]
    image.save(f"{output_dir}/image_{i+1}.png")

print("Synthetic image dataset generated successfully!")

```

```
{"model_id": "abddelb5467c4c8ea53bae31c5f99a9c", "version_major": 2, "version_minor": 0}  
  
{"model_id": "4a808b0cb07e4d9db26d7651fb955158", "version_major": 2, "version_minor": 0}  
  
{"model_id": "ed03fc104681481e89fd3ea07b1184b8", "version_major": 2, "version_minor": 0}  
  
{"model_id": "0d54ba72d7bc411e80ff6a72242a53c4", "version_major": 2, "version_minor": 0}  
  
{"model_id": "075b041ea22b4287b81068f4c3c2d4a3", "version_major": 2, "version_minor": 0}
```

Synthetic image dataset generated successfully!

Display sample output

```
img = Image.open("synthetic_image_dataset/image_1.png")  
plt.imshow(img)  
plt.axis("off")
```

```
(np.float64(-0.5), np.float64(511.5), np.float64(511.5), np.float64(-0.5))
```

