Stable Diffusion XL Turbo Image-to-image

View on GitHub

This example is similar to the Stable Diffusion XL example, but it's a distilled model trained for real-time synthesis and is image-to-image. Learn more about it here.

Input prompt: `dog wizard, gandalf, lord of the rings, detailed, fantasy, cute, adorable, Pixar, Disney, 8k`

Input

Output

Basic setup

```
from io import BytesIO
from pathlib import Path

from modal import App, Image, build, enter, gpu, method
```

Define a container image

```
image = Image.debian_slim().pip_install(
    "Pillow~=10.1.0",
    "diffusers~=0.24.0",
    "transformers~=4.35.2",  # This is needed for `import torch`
    "accelerate~=0.25.0",  # Allows `device_map="auto"``, which allows computation of opti
    "safetensors~=0.4.1",  # Enables safetensor format as opposed to using unsafe pickle f
)

app = App("stable-diffusion-xl-turbo", image=image)

with image.imports():
    import torch
    from diffusers import AutoPipelineForImage2Image
```

```
    from diffusers.utils import load_image
    from huggingface_hub import snapshot_download
    from PIL import Image
```

Load model and run inference

The container lifecycle `@enter` decorator loads the model at startup. Then, we evaluate it in the `inference` function.

To avoid excessive cold-starts, we set the idle timeout to 240 seconds, meaning once a GPU has loaded the model it will stay online for 4 minutes before spinning down. This can be adjusted for cost/experience trade-offs.

```
@app.cls(gpu=gpu.A10G(), container_idle_timeout=240)
class Model:
    @build()
    def download_models(self):
        # Ignore files that we don't need to speed up download time.
        ignore = [
            "*.bin",
            "*.onnx_data",
            "*/diffusion_pytorch_model.safetensors",
        ]

        snapshot_download("stabilityai/sdxl-turbo", ignore_patterns=ignore)

    @enter()
    def enter(self):
        self.pipe = AutoPipelineForImage2Image.from_pretrained(
            "stabilityai/sdxl-turbo",
            torch_dtype=torch.float16,
            variant="fp16",
            device_map="auto",
        )

    @method()
    def inference(self, image_bytes, prompt):
        init_image = load_image(Image.open(BytesIO(image_bytes))).resize(
            (512, 512)
        )
        num_inference_steps = 4
        strength = 0.9
        # "When using SDXL-Turbo for image-to-image generation, make sure that num_inferen
        # See: https://huggingface.co/stabilityai/sdxl-turbo
        assert num_inference_steps * strength >= 1

        image = self.pipe(
            prompt,
```

```
            image=init_image,
            num_inference_steps=num_inference_steps,
            strength=strength,
            guidance_scale=0.0,
        ).images[0]

        byte_stream = BytesIO()
        image.save(byte_stream, format="PNG")
        image_bytes = byte_stream.getvalue()

        return image_bytes


DEFAULT_IMAGE_PATH = Path(__file__).parent / "demo_images/dog.png"


@app.local_entrypoint()
def main(
    image_path=DEFAULT_IMAGE_PATH,
    prompt="dog wizard, gandalf, lord of the rings, detailed, fantasy, cute, adorable, Pix
):
    with open(image_path, "rb") as image_file:
        input_image_bytes = image_file.read()
        output_image_bytes = Model().inference.remote(input_image_bytes, prompt)

    dir = Path("/tmp/stable-diffusion-xl-turbo")
    if not dir.exists():
        dir.mkdir(exist_ok=True, parents=True)

    output_path = dir / "output.png"
    print(f"Saving it to {output_path}")
    with open(output_path, "wb") as f:
        f.write(output_image_bytes)
```

Running the model

We can run the model with different parameters using the following command,

```
modal run stable_diffusion_xl_turbo.py --prompt="harry potter, glasses, wizard" --image-pa
```