Sketch-to-Image

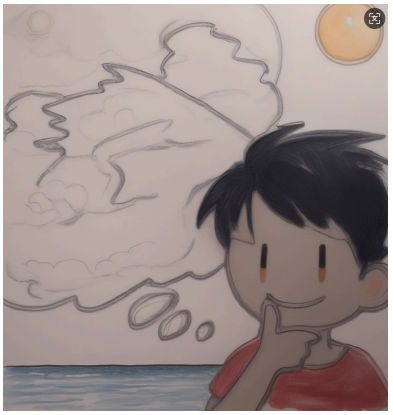
Some models recording:

1. **ControlNet × (ineffective)**
2. **FluxFill × (Need Mask Image)**
3. **Qwen3 √ (effective but need large memory — maybe use web interface..)**

```
import os
from PIL import Image
import torch

from diffusers import QwenImageEditPipeline

pipeline = QwenImageEditPipeline.from_pretrained("Qwen/Qwen-Image-Edit")
print("pipeline loaded")
pipeline.to(torch.bfloat16)
pipeline.to("cuda")
pipeline.set_progress_bar_config(disable=None)
image = Image.open("sketch.png").convert("RGB")
prompt = "Change the sketch into realistic image."
inputs = {
    "image": image,
    "prompt": prompt,
    "generator": torch.manual_seed(0),
    "true_cfg_scale": 4.0,
```

```
    "negative_prompt": " ",
    "num_inference_steps": 50,
}

with torch.inference_mode():
    output = pipeline(**inputs)
    output_image = output.images[0]
    output_image.save("output_image_edit.png")
    print("image saved at", os.path.abspath("output_image_edit.png"))
```

4. **T2I-Adapter √ (can't transform into real photo but can make pic looks more realistic..)**



```
import os
from diffusers import StableDiffusionXLAdapterPipeline, T2IAdapter, EulerA
ncestralDiscreteScheduler, AutoencoderKL
from diffusers.utils import load_image, make_image_grid
from controlnet_aux.pidi import PidiNetDetector
import torch

# load adapter
adapter = T2IAdapter.from_pretrained(
  "TencentARC/t2i-adapter-sketch-sdxl-1.0", torch_dtype=torch.float16, vari
ent="fp16"
).to("cuda")

# load euler_a scheduler
model_id = 'stabilityai/stable-diffusion-xl-base-1.0'
```

```
euler_a = EulerAncestralDiscreteScheduler.from_pretrained(model_id, subf
older="scheduler")
vae=AutoencoderKL.from_pretrained("madebyollin/sdxl-vae-fp16-fix", torc
h_dtype=torch.float16)
pipe = StableDiffusionXLAdapterPipeline.from_pretrained(
    model_id, vae=vae, adapter=adapter, scheduler=euler_a, torch_dtype=to
rch.float16, variant="fp16",
).to("cuda")
pipe.enable_xformers_memory_efficient_attention()

pidinet = PidiNetDetector.from_pretrained("lllyasviel/Annotators").to("cud
a")

# Image
url = "sketch_cartoon.png"
image = load_image(url)
image = pidinet(
  image, detect_resolution=1024, image_resolution=1024, apply_filter=True
)

# generation
prompt = "A boy is thinking his holiday"
negative_prompt = "extra digit, fewer digits, cropped, worst quality, low qu
ality, glitch, deformed, mutated, ugly, disfigured"

gen_images = pipe(
    prompt=prompt,
    negative_prompt=negative_prompt,
    image=image,
    num_inference_steps=30,
    adapter_conditioning_scale=0.9,
    guidance_scale=7.5,
).images[0]
gen_images.save('out_sketch.png')
```