```
!pip install --upgrade diffusers transformers accelerate
!pip install torch torchvision diffusers transformers quickdraw scipy ftfy accelerate xfor
```

```
Collecting diffusers
  Downloading diffusers-0.29.2-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-package
Collecting transformers
  Downloading transformers-4.43.3-py3-none-any.whl.metadata (43 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.7/43.7 kB 2.9 MB/s eta 0:00:00
Requirement already satisfied: accelerate in /usr/local/lib/python3.10/dist-packages
Collecting accelerate
  Downloading accelerate-0.33.0-py3-none-any.whl.metadata (18 kB)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: huggingface-hub>=0.23.2 in /usr/local/lib/python3.10/d
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/di
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.1
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (fro
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl.metadat
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl.metad
Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl.metadat
Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cudnn_cu12-8.9.2.26-py3-none-manylinux1_x86_64.whl.metadata (1.
Collecting nvidia-cublas-cu12==12.1.3.1 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cublas_cu12-12.1.3.1-py3-none-manylinux1_x86_64.whl.metadata (1
Collecting nvidia-cufft-cu12==11.0.2.54 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cufft_cu12-11.0.2.54-py3-none-manylinux1_x86_64.whl.metadata (1
Collecting nvidia-curand-cu12==10.3.2.106 (from torch>=1.10.0->accelerate)
  Using cached nvidia_curand_cu12-10.3.2.106-py3-none-manylinux1_x86_64.whl.metadata
Collecting nvidia-cusolver-cu12==11.4.5.107 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cusolver_cu12-11.4.5.107-py3-none-manylinux1_x86_64.whl.metadat
Collecting nvidia-cusparse-cu12==12.1.0.106 (from torch>=1.10.0->accelerate)
  Using cached nvidia_cusparse_cu12-12.1.0.106-py3-none-manylinux1_x86_64.whl.metadat
Collecting nvidia-nccl-cu12==2.20.5 (from torch>=1.10.0->accelerate)
  Using cached nvidia_nccl_cu12-2.20.5-py3-none-manylinux2014_x86_64.whl.metadata (1.
Collecting nvidia-nvtx-cu12==12.1.105 (from torch>=1.10.0->accelerate)
  Using cached nvidia_nvtx_cu12-12.1.105-py3-none-manylinux1_x86_64.whl.metadata (1.7
Requirement already satisfied: triton==2.3.1 in /usr/local/lib/python3.10/dist-packag
Collecting nvidia-nvjitlink-cu12 (from nvidia-cusolver-cu12==11.4.5.107->torch>=1.10.
  Downloading nvidia_nvjitlink_cu12-12.5.82-py3-none-manylinux2014_x86_64.whl.metadat
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/
```

```
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-p
```

```
!pip install --upgrade torch torchvision
```

```
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (2.4.
Requirement already satisfied: torchvision in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in /usr/local/lib/pyt
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in /usr/local/lib/p
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in /usr/local/lib/pyt
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in /usr/local/lib/python3.
Requirement already satisfied: nvidia-cublas-cu12==12.1.3.1 in /usr/local/lib/python3
Requirement already satisfied: nvidia-cufft-cu12==11.0.2.54 in /usr/local/lib/python3
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in /usr/local/lib/pytho
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in /usr/local/lib/pyt
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in /usr/local/lib/pyt
Requirement already satisfied: nvidia-nccl-cu12==2.20.5 in /usr/local/lib/python3.10/
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in /usr/local/lib/python3.1
Requirement already satisfied: triton==3.0.0 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: nvidia-nvjitlink-cu12 in /usr/local/lib/python3.10/dis
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-p
```

```
import torch
import numpy as np
from PIL import Image, ImageDraw
from quickdraw import QuickDrawData
from diffusers import StableDiffusionImg2ImgPipeline
from torchvision import transforms
```

```
The cache for model files in Transformers v4.22.0 has been updated. Migrating your ol
```

0/0 [00:00<?, ?it/s]

```
WARNING:xformers:WARNING[XFORMERS]: xFormers can't load C++/CUDA extensions. xFormers
    PyTorch 2.3.1+cu121 with CUDA 1201 (you have 2.4.0+cu121)
    Python  3.10.14 (you have 3.10.12)
  Please reinstall xformers (see https://github.com/facebookresearch/xformers#install
  Memory-efficient attention, SwiGLU, sparse and more won't be available.
  Set XFORMERS_MORE_DETAILS=1 for more details
/usr/local/lib/python3.10/dist-packages/xformers/ops/fmha/flash.py:211: FutureWarning
  @torch.library.impl_abstract("xformers_flash::flash_fwd")
/usr/local/lib/python3.10/dist-packages/xformers/ops/fmha/flash.py:344: FutureWarning
  @torch.library.impl_abstract("xformers_flash::flash_bwd")
/usr/local/lib/python3.10/dist-packages/xformers/ops/swiglu_op.py:128: FutureWarning:
  def forward(cls, ctx, x, w1, b1, w2, b2, w3, b3):
/usr/local/lib/python3.10/dist-packages/xformers/ops/swiglu_op.py:149: FutureWarning:
  def backward(cls, ctx, dx5):
```

```
try:
    import google.colab
    IN_COLAB = True
except ImportError:
    IN_COLAB = False


device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
qd = QuickDrawData()



pipe = StableDiffusionImg2ImgPipeline.from_pretrained(
    "runwayml/stable-diffusion-v1-5", torch_dtype=torch.float16, safety_checker=None
).to(device)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarni
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
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 mo
  warnings.warn(
model_index.json: 100%                                  541/541 [00:00<00:00, 34.1kB/s]
Fetching 13 files: 100%                                 13/13 [00:28<00:00,  2.73s/it]
text_encoder/config.json: 100%                          617/617 [00:00<00:00, 32.7kB/s]
tokenizer/vocab.json: 100%                              1.06M/1.06M [00:00<00:00, 5.52MB/s]
(…)ature_extractor/preprocessor_config.json: 100%       342/342 [00:00<00:00, 3.05kB/s]
tokenizer/tokenizer_config.json: 100%                   806/806 [00:00<00:00, 7.41kB/s]
tokenizer/special_tokens_map.json: 100%                 472/472 [00:00<00:00, 3.66kB/s]
tokenizer/merges.txt: 100%                              525k/525k [00:00<00:00, 2.67MB/s]
scheduler/scheduler_config.json: 100%                   308/308 [00:00<00:00, 2.67kB/s]
unet/config.json: 100%                                  743/743 [00:00<00:00, 9.38kB/s]
model.safetensors: 100%                                 492M/492M [00:07<00:00, 115MB/s]
vae/config.json: 100%                                   547/547 [00:00<00:00, 6.14kB/s]
diffusion_pytorch_model.safetensors: 100%               335M/335M [00:06<00:00, 82.8MB/s]
diffusion_pytorch_model.safetensors: 100%               3.44G/3.44G [00:26<00:00, 182MB/s]
Loading pipeline components...: 100%                    6/6 [00:03<00:00,  1.46it/s]
You have disabled the safety checker for <class 'diffusers.pipelines.stable_diffusion
```

```
pipe.enable_attention_slicing()
```

```
def get_sketch(category, size=(512, 512)):
    try:
        sketch = qd.get_drawing(category)
        max_ratio = 5.0
        aspect_ratio = max(sketch.image.width, sketch.image.height) / min(sketch.image.wi
        if aspect_ratio > max_ratio:
            print(f"Warning: Sketch aspect ratio ({aspect_ratio:.2f}) is too extreme, ski
            return None

        # white background image
        image = Image.new('RGB', size, color='white')

        # scaling factor to fit
        scale = min(size[0] / sketch.image.width, size[1] / sketch.image.height)


        scaled_width = int(sketch.image.width * scale)
        scaled_height = int(sketch.image.height * scale)

        # Calculate position to center the sketch
        left = (size[0] - scaled_width) // 2
        top = (size[1] - scaled_height) // 2

        draw = ImageDraw.Draw(image)
        for stroke in sketch.strokes:
            scaled_stroke = [(int(x * scale) + left, int(y * scale) + top) for x, y in st
            draw.line(scaled_stroke, fill='black', width=2)

        return image

    except KeyError:
        print(f"Error: Category '{category}' not found.")
        return None


def get_image_from_sketch(sketch_image, prompt):
    # Preprocess the image
    sketch_image = sketch_image.convert("RGB")

    # Generate image
    image = pipe(prompt=prompt, image=sketch_image, strength=0.75, guidance_scale=7.5).im

    return image


import torch
import torchvision.transforms as transforms
from PIL import Image

def super_resolution(image, scale_factor=2):
    # Convert PIL Image to tensor
    to_tensor = transforms.ToTensor()
    img_tensor = to_tensor(image).unsqueeze(0)

    # Move tensor to the available device (GPU or CPU)
```

```
        # Move tensor to the available device (CPU or GPU)
        device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
        img_tensor = img_tensor.to(device)

        # Upscale the image using bilinear interpolation
        upscaled = torch.nn.functional.interpolate(
            img_tensor,
            scale_factor=scale_factor,
            mode='bilinear',
            align_corners=True
        )

        to_pil = transforms.ToPILImage()
        upscaled_image = to_pil(upscaled.squeeze().cpu().clamp(0, 1))

        return upscaled_image


    category = "house"
    sketch_image = get_sketch(category)
    if sketch_image is not None:
        sketch_image.save(f"sketch_{category}.png")
        print(f"Initial sketch saved as {category}_sketch.png")
        prompt = f"A photorealistic image of a detailed {category}, inspired by a sketch."
        try:
            generated_image = get_image_from_sketch(sketch_image, prompt)
            generated_image.save(f"{category}_pix2pix.png")
            if generated_image:
                enhanced_image = super_resolution(generated_image, scale_factor=2)
                enhanced_image.save(f"enhanced_{category}.png")
                print(f"Enhanced image saved as enhanced_{category}.png")
        except RuntimeError as e:
            if "out of memory" in str(e):
                print("Error: Out of memory. Try reducing image size or complexity.")
            else:
                print(f"An error occurred: {e}")
    else:
        print("Failed to generate sketch.")
```

```
downloading house from https://storage.googleapis.com/quickdraw_dataset/full/binary/h
download complete
loading house drawings
load complete
Initial sketch saved as house_sketch.png
100%                                                 37/37 [00:07<00:00,  3.52it/s]
Enhanced image saved as enhanced_house.png
```

```
#Using ERSGAN
!pip install realesrganimport torch
from realesrgan import RealESRGANer
from realesrgan.archs.srvgg_arch import SRVGGNetCompact
from PIL import Image

model_path = '/opt/anaconda3/envs/diffuser/lib/python3.10/site-packages/realesrgan/weight
netscale = 4

model = SRVGGNetCompact(num_in_ch=3, num_out_ch=3, num_feat=64, num_conv=32, upscale=4, a

upsampler = RealESRGANer(
    scale=netscale,
    model_path=model_path,
    model=model,
    tile=0,
    tile_pad=10,
    pre_pad=0,
    half=True
)

if torch.cuda.is_available():
    upsampler = upsampler.cuda()

def enhance_with_realesrgan(image_path, output_path="enhanced_image.png"):
    try:
        img = Image.open(image_path).convert('RGB')

        with torch.no_grad():
            output, _ = upsampler.enhance(img)

        output.save(output_path)
        print(f"Enhanced image saved to {output_path}")

    except Exception as e:
        print(f"Error during Real-ESRGAN enhancement: {e}")

'''
To run:
enhance_with_realesrgan(image_path)
'''
```

Start coding or generate with AI.