**CODE**

import os

import torch

from diffusers import DiffusionPipeline

import mediapy as media

# Function to display and save the generated images

def display\_and\_save\_images(images, base\_filename="output.jpg"):

    # Determine a unique filename if the base file exists

    new\_filename = base\_filename

    if os.path.exists(base\_filename):

        index = 1

        while True:

            new\_filename = f"output\_{index}.jpg"

            if not os.path.exists(new\_filename):

                break

            index += 1

    # Save the first generated image

    images[0].save(new\_filename)

    print(f"Image saved as: {new\_filename}")

    # Display the images

    media.show\_images(images)

# Load the pre-trained model (e.g., Stable Diffusion)

def load\_model(model\_name="digiplay/Realisian\_v5", use\_gpu=True):

    pipe = DiffusionPipeline.from\_pretrained(

        model\_name,

        torch\_dtype=torch.float16 if use\_gpu else torch.float32,

        safety\_checker=None,

        requires\_safety\_checker=False

    )

    if use\_gpu and torch.cuda.is\_available():

        pipe = pipe.to("cuda")  # Use GPU if available

    return pipe

# Main function to generate images based on a text prompt

def generate\_image(pipe, prompt, seed=4329492846, width=512, height=736, negative\_prompt="", num\_inference\_steps=50):

    generator = torch.Generator().manual\_seed(seed)

    images = pipe(

        prompt=prompt,

        width=width,

        height=height,

        negative\_prompt=negative\_prompt,

        generator=generator,

        output\_type="pil",  # Outputs PIL images

        num\_inference\_steps=num\_inference\_steps  # Reduce the number of inference steps for faster generation

    ).images

    return images

if \_\_name\_\_ == "\_\_main\_\_":

    # Configuration for model and image generation

    model\_name = "digiplay/Realisian\_v5"  # Choose your model

    prompt = "A futuristic cityscape with flying cars and neon lights"  # Input your prompt here

    seed = 4329492846  # Set your desired seed for reproducibility

    width = 512  # Image width

    height = 736  # Image height

    negative\_prompt = "(lowres:2), (bad anatomy:2), (bad hands:2)"  # Define your negative prompts

    # Load the pre-trained model

    print(f"Loading model {model\_name}...")

    model\_pipe = load\_model(model\_name)

    # Generate image based on the provided prompt

    print(f"Generating image for prompt: '{prompt}'...")

    generated\_images = generate\_image(model\_pipe, prompt, seed, width, height, negative\_prompt, num\_inference\_steps=50)

    # Display and save the generated images

    display\_and\_save\_images(generated\_images)

**OUTPUT**

Loading model digiplay/Realisian\_v5...

Loading pipeline components...: 100%

 6/6 [00:25<00:00,  4.26s/it]

Generating image for prompt: 'A futuristic cityscape with flying cars and neon lights'...

****