Stabledfn_mdlreport

August 22, 2023

[1]:
```
! git clone https://github.com/Jawakar-7/StableDiffusion-Models.git
```

```
Cloning into 'StableDiffusion-Models'…
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 33 (delta 5), reused 12 (delta 0), pack-reused 0
Receiving objects: 100% (33/33), 28.85 MiB | 13.44 MiB/s, done.
Resolving deltas: 100% (5/5), done.
```

[1]:
```
import warnings
warnings.filterwarnings("ignore")
```

0.1 Stable Diffusion :

Stable Diffusion is a text-to-image latent diffusion model created by the researchers and engineers from CompVis, Stability AI and LAION. It's trained on 512x512 images from a subset of the LAION-5B database. This model uses a frozen CLIP ViT-L/14 text encoder to condition the model on text prompts. With its 860M UNet and 123M text encoder, the model is relatively lightweight and can run on many consumer GPUs. Here the first Model I'm trying to use is a ###'CompVis/stable-diffusion-v1-4" model

[2]:
```
!nvidia-smi
#to check if the model is connected with a GPU runtime
```

```
Tue Aug 22 04:06:17 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.105.17   Driver Version: 525.105.17   CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   50C    P8    10W /  70W |      0MiB / 15360MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
```

```
| Processes:                                                                 |
|  GPU   GI   CI        PID   Type   Process name                 GPU Memory |
|        ID   ID                                                  Usage      |
|============================================================================|
|  No running processes found                                                |
+----------------------------------------------------------------------------+
```

0.1.1 Install Dependencies

[3]:
```
!pip install diffusers
!pip install transformers scipy ftfy accelerate
```

```
Collecting diffusers
  Downloading diffusers-0.20.0-py3-none-any.whl (1.3 MB)
                             1.3/1.3 MB
10.2 MB/s eta 0:00:00
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.10/dist-packages (from diffusers) (6.8.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from diffusers) (3.12.2)
Collecting huggingface-hub>=0.13.2 (from diffusers)
  Downloading huggingface_hub-0.16.4-py3-none-any.whl (268 kB)
                            268.8/268.8 kB
13.3 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-
packages (from diffusers) (1.23.5)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from diffusers) (2023.6.3)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from diffusers) (2.31.0)
Collecting safetensors>=0.3.1 (from diffusers)
  Downloading
safetensors-0.3.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1.3 MB)
                             1.3/1.3 MB
18.1 MB/s eta 0:00:00
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-
packages (from diffusers) (9.4.0)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages
(from huggingface-hub>=0.13.2->diffusers) (2023.6.0)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub>=0.13.2->diffusers) (4.66.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub>=0.13.2->diffusers) (6.0.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub>=0.13.2->diffusers) (4.7.1)
Requirement already satisfied: packaging>=20.9 in
```

```
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub>=0.13.2->diffusers) (23.1)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-
packages (from importlib-metadata->diffusers) (3.16.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->diffusers) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->diffusers) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->diffusers) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->diffusers) (2023.7.22)
Installing collected packages: safetensors, huggingface-hub, diffusers
Successfully installed diffusers-0.20.0 huggingface-hub-0.16.4 safetensors-0.3.2
Collecting transformers
  Downloading transformers-4.31.0-py3-none-any.whl (7.4 MB)
                           7.4/7.4 MB
20.2 MB/s eta 0:00:00
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-
packages (1.10.1)
Collecting ftfy
  Downloading ftfy-6.1.1-py3-none-any.whl (53 kB)
                           53.1/53.1 kB
7.1 MB/s eta 0:00:00
Collecting accelerate
  Downloading accelerate-0.21.0-py3-none-any.whl (244 kB)
                          244.2/244.2 kB
30.6 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from transformers) (3.12.2)
Requirement already satisfied: huggingface-hub<1.0,>=0.14.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.16.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-
packages (from transformers) (1.23.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from transformers) (23.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from transformers) (2023.6.3)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from transformers) (2.31.0)
Collecting tokenizers!=0.11.3,<0.14,>=0.11.1 (from transformers)
  Downloading
tokenizers-0.13.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(7.8 MB)
                           7.8/7.8 MB
55.1 MB/s eta 0:00:00
```

```
Requirement already satisfied: safetensors>=0.3.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.3.2)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-
packages (from transformers) (4.66.1)
Requirement already satisfied: wcwidth>=0.2.5 in /usr/local/lib/python3.10/dist-
packages (from ftfy) (0.2.6)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages
(from accelerate) (5.9.5)
Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-
packages (from accelerate) (2.0.1+cu118)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages
(from huggingface-hub<1.0,>=0.14.1->transformers) (2023.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.14.1->transformers) (4.7.1)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages
(from torch>=1.10.0->accelerate) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-
packages (from torch>=1.10.0->accelerate) (3.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages
(from torch>=1.10.0->accelerate) (3.1.2)
Requirement already satisfied: triton==2.0.0 in /usr/local/lib/python3.10/dist-
packages (from torch>=1.10.0->accelerate) (2.0.0)
Requirement already satisfied: cmake in /usr/local/lib/python3.10/dist-packages
(from triton==2.0.0->torch>=1.10.0->accelerate) (3.27.2)
Requirement already satisfied: lit in /usr/local/lib/python3.10/dist-packages
(from triton==2.0.0->torch>=1.10.0->accelerate) (16.0.6)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->transformers) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(2023.7.22)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.10.0->accelerate)
(2.1.3)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-
packages (from sympy->torch>=1.10.0->accelerate) (1.3.0)
Installing collected packages: tokenizers, ftfy, transformers, accelerate
Successfully installed accelerate-0.21.0 ftfy-6.1.1 tokenizers-0.13.3
transformers-4.31.0
```

0.1.2 Pipeline

StableDiffusionPipeline is an end-to-end inference pipeline that you can use to generate images from text with just a few lines of code.

First, we load the pre-trained weights of all components of the model

[4]:
```
import torch
from diffusers import StableDiffusionPipeline
```

0.1.3 Importing pretrained CompVis/stable-diffusion-v1-4 Model

[6]:
```
mypipe=StableDiffusionPipeline.from_pretrained("CompVis/
  ↪stable-diffusion-v1-4",torch_dtype=torch.float16)
```

```
Downloading (…)ain/model_index.json:   0%|          | 0.00/541 [00:00<?, ?B/s]
Fetching 16 files:   0%|          | 0/16 [00:00<?, ?it/s]
Downloading (…)_checker/config.json:   0%|          | 0.00/4.56k [00:00<?, ?B/s]
Downloading (…)cheduler_config.json:   0%|          | 0.00/313 [00:00<?, ?B/s]
Downloading (…)okenizer_config.json:   0%|          | 0.00/806 [00:00<?, ?B/s]
Downloading model.safetensors:   0%|          | 0.00/492M [00:00<?, ?B/s]
Downloading (…)nfig-checkpoint.json:   0%|          | 0.00/209 [00:00<?, ?B/s]
Downloading (…)tokenizer/merges.txt:   0%|          | 0.00/525k [00:00<?, ?B/s]
Downloading model.safetensors:   0%|          | 0.00/1.22G [00:00<?, ?B/s]
Downloading (…)_encoder/config.json:   0%|          | 0.00/592 [00:00<?, ?B/s]
Downloading (…)rocessor_config.json:   0%|          | 0.00/342 [00:00<?, ?B/s]
Downloading (…)cial_tokens_map.json:   0%|          | 0.00/472 [00:00<?, ?B/s]
Downloading (…)tokenizer/vocab.json:   0%|          | 0.00/1.06M [00:00<?, ?B/s]
Downloading (…)69ce/vae/config.json:   0%|          | 0.00/551 [00:00<?, ?B/s]
Downloading (…)9ce/unet/config.json:   0%|          | 0.00/743 [00:00<?, ?B/s]
Downloading (…)ch_model.safetensors:   0%|          | 0.00/3.44G [00:00<?, ?B/s]
Downloading (…)ch_model.safetensors:   0%|          | 0.00/335M [00:00<?, ?B/s]
Loading pipeline components…:   0%|          | 0/7 [00:00<?, ?it/s]
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["id2label"]` will be overriden.
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["bos_token_id"]` will be overriden.
```

```
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["eos_token_id"]` will be overriden.
```

[7]:
```
mypipe=mypipe.to("cuda")
```

Now that we have moved our model to GPU we will try and create images using Stable Diffusion API

[8]:
```
prompt="Batman VS superman"
```

[9]:
```
img1=mypipe(prompt).images[0]
```

```
  0%|          | 0/50 [00:00<?, ?it/s]
```

```
Potential NSFW content was detected in one or more images. A black image will be
returned instead. Try again with a different prompt and/or seed.
```

Running the above cell multiple times will give you a different image every time. If you want deterministic output you can pass a random seed to the pipeline. Every time you use the same seed you'll have the same image result. and also we can change the number of inference steps using the num_inference_steps argument. In general, results are better the more steps you use. Stable Diffusion, being one of the latest models, works great with a relatively small number of steps, so we recommend to use the default of 50.

0.1.4 Since we are trying to use a Stable diffusion model for creating comic images, it is better if we have the same seed , but we will not know for sure so im creating two functions to check the relavency

[11]:
```
import torch
generator = torch.Generator("cuda").manual_seed(1024)
image = mypipe(prompt, num_inference_steps=15, generator=generator).images[0]
image
```

```
  0%|          | 0/15 [00:00<?, ?it/s]
```

[11]:

[ ]:
```
import torch

generator = torch.Generator("cuda").manual_seed(1024)

image = mypipe(prompt, num_inference_steps=95, generator=generator).images[0]

image
```

Here we can see that the quality of images gets increased for each number of inference steps and since we have set a seed value the same image will be reapeated no matter how many time we run the same prompt .

#####The other parameter in the pipeline call is the image size , defaultly it will be creating square images , but we can manually specify the size of images Make sure height and width are

both multiples of 8. Going below 512 might result in lower quality images.

[13]:
```
h=1024
w=1024
image2=mypipe(prompt,h,w).images[0]
```

```
  0%|          | 0/50 [00:00<?, ?it/s]
```

[14]:
```
image2
```

[14]:

0.2 Now that we have seen all the basics , i will try implementing different prompts of our work

0.2.1 Here I am adding prompts created by me, and references from several websites and compare the results

https://openart.ai/discovery?q=comic&searchType=community-only&searchModel=sd

https://arc.tencent.com/en/ai-demos/imgRestore

https://publicprompts.art/comic-art/

[6]:
```
p1="A photorealistic image of Batman and Superman fighting in a city. Batman is␣
 ↪wearing his black Batsuit and Superman is wearing his red and blue Superman␣
 ↪suit. They are both in mid-air, facing each other. The city is in ruins␣
 ↪around them, with buildings and cars flying. The sky is dark and stormy."
```

[7]:
```
p2="""Batman vs Superman

Batman: Im Batman.
Superman: Im Superman.
City: Oh no.
This meme is funny because it is a play on the classic superhero trope of two␣
 ↪powerful characters fighting each other. The city is the one who is␣
 ↪suffering the most in this situation, as it is being destroyed by the fight.
 ↪"""
```

[8]:
```
p3="A photorealistic image of Batman and Superman fighting in a city. Batman is␣
 ↪wearing his black Batsuit and Superman is wearing his red and blue Superman␣
 ↪suit. They are both in mid-air, facing each other. The city is in ruins␣
 ↪around them, with buildings and cars flying. The sky is dark and stormy."
```

[9]:
```
p4='Retro comic style artwork, highly detailed James Bond, comic book cover,␣
 ↪symmetrical, vibrant'
```

[10]:
```
p5="Vogue Panda animal, a portrait hipster rapper, atompunk style, highly␣
 ↪detailed and intricated, concept art, retro colors, Art Station, octane␣
 ↪render, unreal engine, 8k, trending on artstation, Gorgeous"
```

[20]:
```
# #to create a grid of images
# from PIL import Image

# def image_grid(imgs, rows, cols):
#     assert len(imgs) == rows*cols

#     w, h = imgs[0].size
#     grid = Image.new('RGB', size=(cols*w, rows*h))
#     grid_w, grid_h = grid.size

#     for i, img in enumerate(imgs):
```

```
#         grid.paste(img, box=(i%cols*w, i//cols*h))
#     return grid
```

[13]:
```
#Defining model parameters
h=1024
w=720
noof_ininference=500
```

[ ]:
```
c_img1=mypipe(p1,generator=generator,height=h,width=w,num_inference_steps=noof_ininference).
↪images[0]
```

[23]:
```
c_img1
```

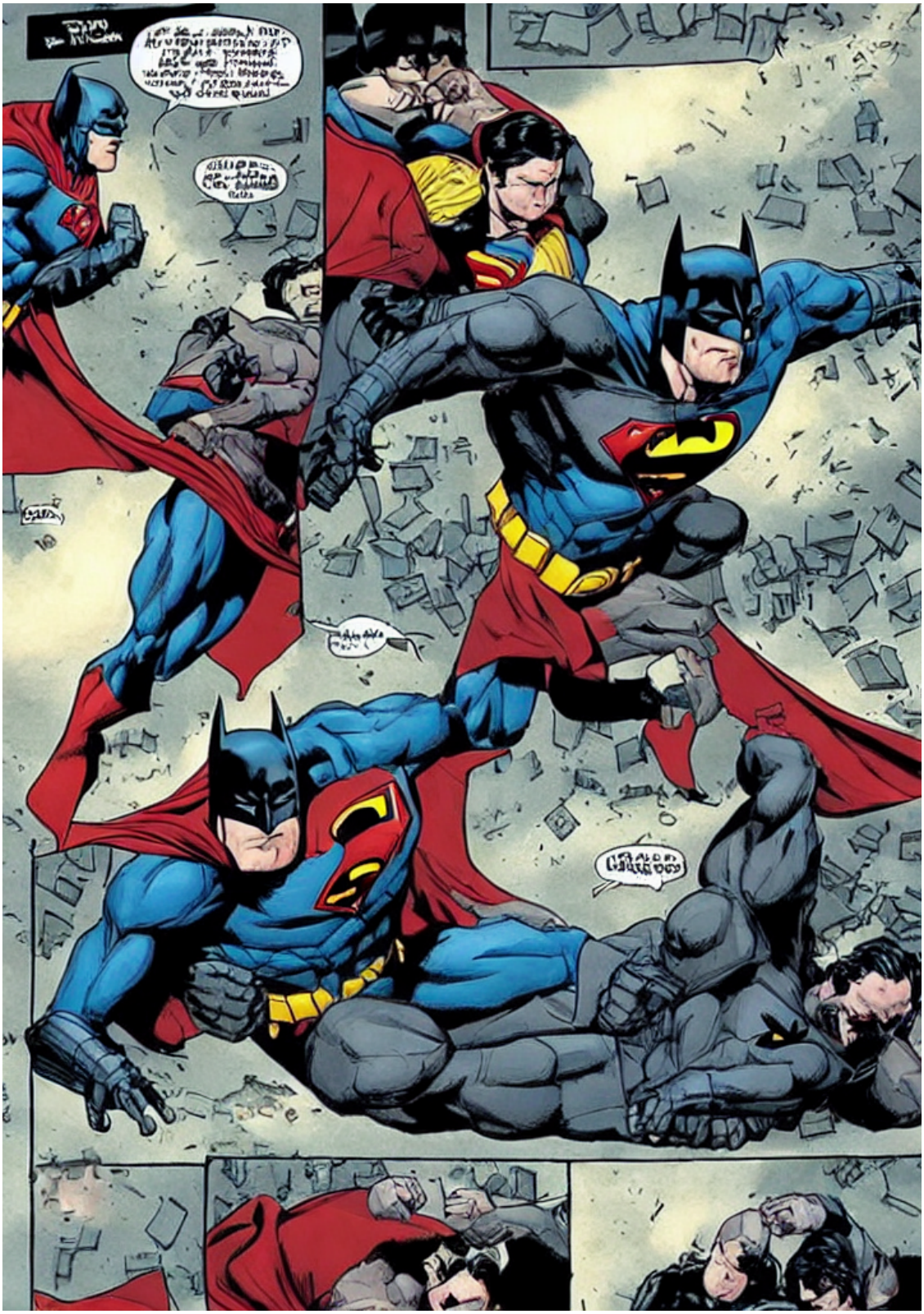
[23]:

[24]:
```
c_img2=mypipe(p2,generator=generator,height=h,width=w,num_inference_steps=500).
↪images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[25]:
```
c_img2
```

[25]:



[26]:
```
c_img3=mypipe(p3,generator=generator,height=h,width=w,num_inference_steps=500).
↪images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[27]:
```
c_img3
```

[27]:

[29]:
```
c_img4=mypipe(p4,generator=generator,height=h,width=w,num_inference_steps=500).
↪images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[30]:
```
c_img4
```

[30]:

[31]:
```
c_img5=mypipe(p5,generator=generator,height=h,width=w,num_inference_steps=500).
↪images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[32]:
```
c_img5
```

[32]:

[ ]:
```
import os
os.kill(os.getpid(), 9)
#since the Gpu might get overloaded we are using this function to kill the␣
 ↪runtime and start again
```

0.3 runwayml/stable-diffusion-v1-5

Stable Diffusion is a latent text-to-image diffusion model capable of generating photo-realistic images given any text input

[2]:
```
p1="A photorealistic image of Batman and Superman fighting in a city. Batman is␣
 ↪wearing his black Batsuit and Superman is wearing his red and blue Superman␣
 ↪suit. They are both in mid-air, facing each other. The city is in ruins␣
 ↪around them, with buildings and cars flying. The sky is dark and stormy."
p2="""Batman vs Superman

Batman: Im Batman.
Superman: Im Superman.
City: Oh no.
This meme is funny because it is a play on the classic superhero trope of two␣
 ↪powerful characters fighting each other. The city is the one who is␣
 ↪suffering the most in this situation, as it is being destroyed by the fight.
 ↪"""
p3="A photorealistic image of Batman and Superman fighting in a city. Batman is␣
 ↪wearing his black Batsuit and Superman is wearing his red and blue Superman␣
 ↪suit. They are both in mid-air, facing each other. The city is in ruins␣
 ↪around them, with buildings and cars flying. The sky is dark and stormy."
p4='Retro comic style artwork, highly detailed James Bond, comic book cover,␣
 ↪symmetrical, vibrant'
p5="Vogue Panda animal, a portrait hipster rapper, atompunk style, highly␣
 ↪detailed and intricated, concept art, retro colors, Art Station, octane␣
 ↪render, unreal engine, 8k, trending on artstation, Gorgeous"
```

[11]:
```
from diffusers import StableDiffusionPipeline

model_id = "runwayml/stable-diffusion-v1-5"
runpipe = StableDiffusionPipeline.from_pretrained(model_id, torch_dtype=torch.
 ↪float16, revision="fp16")
runpipe = runpipe.to("cuda")
```

```
Downloading (…)p16/model_index.json:   0%|          | 0.00/543 [00:00<?, ?B/s]

safety_checker/model.safetensors not found

Fetching 15 files:   0%|          | 0/15 [00:00<?, ?it/s]

Downloading (…)_checker/config.json:   0%|          | 0.00/4.70k [00:00<?, ?B/s]

Downloading (…)cheduler_config.json:   0%|          | 0.00/307 [00:00<?, ?B/s]
```

```
Downloading (…)_encoder/config.json:   0%|          | 0.00/636 [00:00<?, ?B/s]
Downloading (…)cial_tokens_map.json:   0%|          | 0.00/472 [00:00<?, ?B/s]
Downloading (…)tokenizer/merges.txt:   0%|          | 0.00/525k [00:00<?, ?B/s]
Downloading (…)rocessor_config.json:   0%|          | 0.00/342 [00:00<?, ?B/s]
Downloading (…)okenizer_config.json:   0%|          | 0.00/822 [00:00<?, ?B/s]
Downloading (…)tokenizer/vocab.json:   0%|          | 0.00/1.06M [00:00<?, ?B/s]
Downloading (…)9cc2/vae/config.json:   0%|          | 0.00/609 [00:00<?, ?B/s]
Downloading (…)cc2/unet/config.json:   0%|          | 0.00/806 [00:00<?, ?B/s]
Downloading pytorch_model.bin:   0%|          | 0.00/246M [00:00<?, ?B/s]
Downloading pytorch_model.bin:   0%|          | 0.00/608M [00:00<?, ?B/s]
Downloading (…)on_pytorch_model.bin:   0%|          | 0.00/1.72G [00:00<?, ?B/s]
Downloading (…)on_pytorch_model.bin:   0%|          | 0.00/167M [00:00<?, ?B/s]
Loading pipeline components…:   0%|          | 0/7 [00:00<?, ?it/s]
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["id2label"]` will be overriden.
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["bos_token_id"]` will be overriden.
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["eos_token_id"]` will be overriden.
```

[15]:
```
rimg1=runpipe(p1,generator=generator,height=h,width=w,num_inference_steps=200).
↪images[0]
```

```
  0%|          | 0/200 [00:00<?, ?it/s]
```

[16]:
```
rimg1
```

[16]:

[17]:
```
rimg2=runpipe(p2,generator=generator,height=h,width=w,num_inference_steps=500).
↪images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[18]:
```
rimg2
```

[18]:

[19]:
```
rimg3=runpipe(p3,generator=generator,height=h,width=w,num_inference_steps=500).
↪images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[20]:
```
rimg3
```

[20]:

[21]:
```
rimg4=runpipe(p4,generator=generator,height=h,width=w,num_inference_steps=500).
↪images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[22]:
```
rimg4
```

[22]:

[23]:
```
rimg5=runpipe(p5,generator=generator,height=h,width=w,num_inference_steps=500).
↪images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[24]:
```
rimg5
```

[24]:

###Inpainting with Stable Diffusion using "runwayml/stable-diffusion-inpainting" model This is just a image upscaling model

[25]:
```
from diffusers import StableDiffusionInpaintPipeline

pipe3 = StableDiffusionInpaintPipeline.from_pretrained(
    "runwayml/stable-diffusion-inpainting",
    revision="fp16",
    torch_dtype=torch.float16,
)
```

```
Downloading (…)p16/model_index.json:   0%|          | 0.00/550 [00:00<?, ?B/s]

safety_checker/model.safetensors not found

Fetching 15 files:   0%|          | 0/15 [00:00<?, ?it/s]

Downloading (…)cial_tokens_map.json:   0%|          | 0.00/472 [00:00<?, ?B/s]

Downloading (…)_checker/config.json:   0%|          | 0.00/4.75k [00:00<?, ?B/s]

Downloading (…)_encoder/config.json:   0%|          | 0.00/635 [00:00<?, ?B/s]

Downloading (…)rocessor_config.json:   0%|          | 0.00/342 [00:00<?, ?B/s]

Downloading (…)tokenizer/merges.txt:   0%|          | 0.00/525k [00:00<?, ?B/s]

Downloading (…)cheduler_config.json:   0%|          | 0.00/287 [00:00<?, ?B/s]

Downloading (…)tokenizer/vocab.json:   0%|          | 0.00/1.06M [00:00<?, ?B/s]

Downloading (…)okenizer_config.json:   0%|          | 0.00/821 [00:00<?, ?B/s]

Downloading (…)6e5/unet/config.json:   0%|          | 0.00/810 [00:00<?, ?B/s]

Downloading (…)66e5/vae/config.json:   0%|          | 0.00/613 [00:00<?, ?B/s]

Downloading pytorch_model.bin:   0%|          | 0.00/608M [00:00<?, ?B/s]

Downloading pytorch_model.bin:   0%|          | 0.00/246M [00:00<?, ?B/s]

Downloading (…)on_pytorch_model.bin:   0%|          | 0.00/167M [00:00<?, ?B/s]

Downloading (…)on_pytorch_model.bin:   0%|          | 0.00/1.72G [00:00<?, ?B/s]

Loading pipeline components…:   0%|          | 0/7 [00:00<?, ?it/s]

`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["id2label"]` will be overriden.
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["bos_token_id"]` will be overriden.
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["eos_token_id"]` will be overriden.
```

[26]:
```
pipe3 = pipe3.to("cuda")
```

[27]:
```
from PIL import Image

# creating a object
im = Image.open(r"/content/img.jpg")
im.show()
```

[28]:
```
p3img1=pipe3(p1,rimg1,im,num_inference_steps=500).images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[29]:
```
p3img1
```

[29]:

[30]:
```
p3img2=pipe3(p2,rimg2,im,num_inference_steps=500).images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[ ]:
```
# import os
# os.kill(os.getpid(), 9)
# #since the Gpu might get overloaded we are using this function to kill the␣
 ↪runtime and start again
```

##ogkalu/Comic-Diffusion Trained on 6 styles at once, it allows anyone to create unique but consistent styles by mixing any number of the tokens. Even changing the order of the same list influences results so there's a lot to experiment with here. This was created so anyone could create their comic projects with ease and flexibility. It is the culmination of all my experimentation with dreambooth thus far.

[1]:
```
p1="A photorealistic image of Batman and Superman fighting in a city. Batman is␣
 ↪wearing his black Batsuit and Superman is wearing his red and blue Superman␣
 ↪suit. They are both in mid-air, facing each other. The city is in ruins␣
 ↪around them, with buildings and cars flying. The sky is dark and stormy."
p2="""Batman vs Superman

Batman: Im Batman.
Superman: Im Superman.
City: Oh no.
This meme is funny because it is a play on the classic superhero trope of two␣
 ↪powerful characters fighting each other. The city is the one who is␣
 ↪suffering the most in this situation, as it is being destroyed by the fight.
 ↪"""
p3="A photorealistic image of Batman and Superman fighting in a city. Batman is␣
 ↪wearing his black Batsuit and Superman is wearing his red and blue Superman␣
 ↪suit. They are both in mid-air, facing each other. The city is in ruins␣
 ↪around them, with buildings and cars flying. The sky is dark and stormy."
p4='Retro comic style artwork, highly detailed James Bond, comic book cover,␣
 ↪symmetrical, vibrant'
p5="Vogue Panda animal, a portrait hipster rapper, atompunk style, highly␣
 ↪detailed and intricated, concept art, retro colors, Art Station, octane␣
 ↪render, unreal engine, 8k, trending on artstation, Gorgeous"
```

[2]:
```
from diffusers import DiffusionPipeline

comicpipeline = DiffusionPipeline.from_pretrained("ogkalu/Comic-Diffusion")
```

```
Downloading (…)ain/model_index.json:   0%|          | 0.00/546 [00:00<?, ?B/s]

text_encoder/model.safetensors not found

Fetching 16 files:   0%|          | 0/16 [00:00<?, ?it/s]

Downloading (…)_checker/config.json:   0%|          | 0.00/4.84k [00:00<?, ?B/s]

Downloading (…)cheduler_config.json:   0%|          | 0.00/313 [00:00<?, ?B/s]

Downloading (…)tokenizer/merges.txt:   0%|          | 0.00/525k [00:00<?, ?B/s]
```

```
Downloading (…)_encoder/config.json:   0%|          | 0.00/612 [00:00<?, ?B/s]
Downloading (…)rocessor_config.json:   0%|          | 0.00/342 [00:00<?, ?B/s]
Downloading (…)cial_tokens_map.json:   0%|          | 0.00/472 [00:00<?, ?B/s]
Downloading (…)okenizer_config.json:   0%|          | 0.00/806 [00:00<?, ?B/s]
Downloading (…)9a8/unet/config.json:   0%|          | 0.00/748 [00:00<?, ?B/s]
Downloading (…)tokenizer/vocab.json:   0%|          | 0.00/1.06M [00:00<?, ?B/s]
Downloading (…)39a8/vae/config.json:   0%|          | 0.00/581 [00:00<?, ?B/s]
Downloading pytorch_model.bin:   0%|          | 0.00/492M [00:00<?, ?B/s]
Downloading (…)sion Â· Hugging Face:   0%|          | 0.00/15.7M [00:00<?, ?B/s]
Downloading (…)on_pytorch_model.bin:   0%|          | 0.00/3.44G [00:00<?, ?B/s]
Downloading pytorch_model.bin:   0%|          | 0.00/1.22G [00:00<?, ?B/s]
Downloading (…)on_pytorch_model.bin:   0%|          | 0.00/335M [00:00<?, ?B/s]
Loading pipeline components…:   0%|          | 0/7 [00:00<?, ?it/s]
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["id2label"]` will be overriden.
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["bos_token_id"]` will be overriden.
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["eos_token_id"]` will be overriden.
```

[3]:
```
comicpipeline = comicpipeline.to("cuda")
```

[4]:
```
comic1=comicpipeline(p1,num_inference_steps=300).images[0]
```

```
  0%|          | 0/300 [00:00<?, ?it/s]
```

[5]:
```
comic1
```

[5]:

[6]:
```
comic2=comicpipeline(p2,num_inference_steps=300).images[0]
```

```
  0%|          | 0/300 [00:00<?, ?it/s]
```

[7]:
```
comic2
```

[7]:

[8]:
```
comic3=comicpipeline(p3,num_inference_steps=300).images[0]
```

```
  0%|          | 0/300 [00:00<?, ?it/s]
```

[9]:
```
comic3
```

[9]:

[10]:
```
comic4=comicpipeline(p4,num_inference_steps=300).images[0]
```

```
  0%|          | 0/300 [00:00<?, ?it/s]
```

[11]:
```
comic4
```

[11]:

[12]:
```
comic5=comicpipeline(p5,num_inference_steps=300).images[0]
```

```
  0%|          | 0/300 [00:00<?, ?it/s]
```

[13]:
```
comic5
```

[13]:

I have used prompts from https://publicprompts.art/comic-art/ website

#PublicPrompts/borderlands

[14]:
```
from diffusers import DiffusionPipeline

borline = DiffusionPipeline.from_pretrained("PublicPrompts/borderlands")
```

```
Downloading (…)ain/model_index.json:   0%|          | 0.00/546 [00:00<?, ?B/s]
Fetching 15 files:   0%|          | 0/15 [00:00<?, ?it/s]
Downloading (…)rocessor_config.json:   0%|          | 0.00/342 [00:00<?, ?B/s]
Downloading (…)_encoder/config.json:   0%|          | 0.00/612 [00:00<?, ?B/s]
```

```
Downloading (…)_checker/config.json:   0%|          | 0.00/4.84k [00:00<?, ?B/s]
Downloading (…)cheduler_config.json:   0%|          | 0.00/313 [00:00<?, ?B/s]
Downloading (…)tokenizer/merges.txt:   0%|          | 0.00/525k [00:00<?, ?B/s]
Downloading (…)cial_tokens_map.json:   0%|          | 0.00/472 [00:00<?, ?B/s]
Downloading (…)okenizer_config.json:   0%|          | 0.00/806 [00:00<?, ?B/s]
Downloading (…)tokenizer/vocab.json:   0%|          | 0.00/1.06M [00:00<?, ?B/s]
Downloading (…)fbab/vae/config.json:   0%|          | 0.00/581 [00:00<?, ?B/s]
Downloading (…)bab/unet/config.json:   0%|          | 0.00/748 [00:00<?, ?B/s]
Downloading model.safetensors:   0%|          | 0.00/492M [00:00<?, ?B/s]
Downloading model.safetensors:   0%|          | 0.00/1.22G [00:00<?, ?B/s]
Downloading (…)ch_model.safetensors:   0%|          | 0.00/3.44G [00:00<?, ?B/s]
Downloading (…)ch_model.safetensors:   0%|          | 0.00/335M [00:00<?, ?B/s]
Loading pipeline components…:   0%|          | 0/7 [00:00<?, ?it/s]
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["id2label"]` will be overriden.
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["bos_token_id"]` will be overriden.
`text_config_dict` is provided which will be used to initialize
`CLIPTextConfig`. The value `text_config["eos_token_id"]` will be overriden.
```

[15]:
```
borline = borline.to("cuda")
```

[16]:
```
bore1=borline(p1,num_inference_steps=500).images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[17]:
```
bore1
```

[17]:

[18]:
```
bore2=borline(p2,num_inference_steps=500).images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[19]:
```
bore2
```

[19]:

[20]:
```
bore3=borline(p3,num_inference_steps=500).images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[21]:
```
bore3
```

[21]:

[22]:
```
bore4=borline(p4,num_inference_steps=500).images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[23]:
```
bore4
```

[23]:

[24]:
```
bore5=borline(p5,num_inference_steps=500).images[0]
```

```
  0%|          | 0/500 [00:00<?, ?it/s]
```

[26]:
```
bore5
```

[26]:

1 Conclusion :

In this model, we have used the famous “CompVis/stable-diffusion-v1-4”,“runwayml/stable-diffusion-v1-5”,“ogkalu/Comic-Diffusion”,“PublicPrompts/borderlands” pretrained model pipeline and an inpainting diffusion model to create our images increase their accuracy.

We have seen in detail, how to create a api reference from huggingface and download the pretrained pipeline to create images from text.And have used five distinct prompts , all related to Comic books to create images.

1.0.1 Comparision:

Of all the models the commic-diffusion model and boderlands model stood out as they are specifically designed for creating comic images. The image accuracy can be increased by using better prompts and proper hyperparameter tuning.

##References: https://huggingface.co/ogkalu/Comic-Diffusion

https://huggingface.co/docs/api-inference/

https://huggingface.co/PublicPrompts/borderlands/blob/main/README.md

https://colab.research.google.com/github/huggingface/notebooks/blob/main/diffusers/stable_diffusion.ipynb#sc

https://towardsdatascience.com/stable-diffusion-using-hugging-face-501d8dbdd8

[ ]: