



# **DIVYA DHARSHINI.S**

FINAL PROJECT



# **GENERATING REALISTIC IMAGES**



# AGENDA

**1** PROBLEM STATEMENT

**2** OVERVIEW

**3** END USERS

**4** SOLUTION

**5** MODELLING

**6** RESULTS



# PROBLEM STATEMENT


GENERATING REALISTIC IMAGES  
USING GENERATIVE AI





# OVERVIEW

USING GENERATIVE AI, REALISTIC PICTURE GENERATION ENTAILS CREATING IMAGES THAT CLOSELY MIMIC REAL-WORLD PHOTOS OR ARTWORK BY UTILIZING DEEP LEARNING ARCHITECTURES, VARIATIONAL AUTOENCODERS (VAES), AND GENERATIVE ADVERSARIAL NETWORKS (GANS). APPLICATIONS FOR THIS TASK CAN BE FOUND IN MANY OTHER DISCIPLINES.





# **END USERS**


END USERS OF GENERATING REALISTIC IMAGES USING GENERATIVE AI ENCOMPASS VARIOUS INDUSTRIES AND APPLICATIONS.

- **ENTERTAINMENT AND MEDIA**
  - **ADVERTISING AND MARKETING**
  - **EDUCATION AND RESEARCH**
  - **SIMULATION AND TRAINING**
  - **MEDICAL IMAGING**
  - **E-COMMERCE AND RETAIL**
- 



# **SOLUTION**

THE DEVELOPMENT OF GENERATIVE AI ALGORITHMS HAS MADE IT POSSIBLE TO PRODUCE REALISTIC VISUALS. IT IS STILL DIFFICULT TO PRODUCE VARIED, HIGH-QUALITY IMAGES THAT CLOSELY MIMIC ACTUAL PHOTOS OR ARTWORK. THE GOAL OF THIS PROJECT IS TO CREATE GENERATIVE AI-BASED REALISTIC PICTURE GENERATION THAT CAN PRODUCE IMAGES WITH HIGH FIDELITY, DIVERSITY, AND COHERENCE IN A VARIETY OF FIELDS.





# MODELLING

- SELECT A PRE-TRAINED MODEL
- FINE-TUNE THE MODEL
- EVALUATE THE MODEL
- GENERATE IMAGES
- DEPLOYMENT AND APPLICATION
- THE IMAGE IS GENERATED



# DEPLOYMENT

```
Diffusers is a hugging face page for using diffusion models from huggingface hub
!pip install diffusers transformers

Collecting diffusers
  Downloading diffusers-0.27.2-py3-none-any.whl (2.0 MB)
    Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (from diffusers) (4.38.2)
    Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.10/dist-packages (from diffusers) (7.1.0)
    Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from diffusers) (3.13.3)
    Requirement already satisfied: huggingface-hub<0.20.2 in /usr/local/lib/python3.10/dist-packages (from diffusers) (0.20.3)
    Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from diffusers) (1.25.2)
    Requirement already satisfied: regex<2019.12.17 in /usr/local/lib/python3.10/dist-packages (from diffusers) (2023.12.25)
    Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from diffusers) (2.31.0)
    Requirement already satisfied: safetensors<0.3.1 in /usr/local/lib/python3.10/dist-packages (from diffusers) (0.4.2)
    Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from diffusers) (9.4.0)
    Requirement already satisfied: packaging>20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.0)
    Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
    Requirement already satisfied: tokenizers<0.19,>=0.14 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.15.2)
    Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.2)
    Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<0.20.2>-diffusers) (4.10.0)
    Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata>-diffusers) (3.18.1)
    Requirement already satisfied: charset-normalizer<=2 in /usr/local/lib/python3.10/dist-packages (from requests>-diffusers) (3.3.2)
    Requirement already satisfied: idna<=,>2.5 in /usr/local/lib/python3.10/dist-packages (from requests>-diffusers) (3.0)
    Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>-diffusers) (2.0.7)
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>-diffusers) (2024.2.2)
Installing collected packages: diffusers
Successfully installed diffusers-0.27.2

[ ] from diffusers import StableDiffusionPipeline
import matplotlib.pyplot as plt
```

```
from diffusers import StableDiffusionPipeline
import matplotlib.pyplot as plt
import torch

The cache for model files in Transformers v4.22.0 has been updated. Migrating your old cache. This is a one-time only operation. You can interrupt this and resume the migration later c
00 [00:00< 78s]

[ ] !pip show torch

Name: torch
Version: 2.2.1+cu121
Summary: Tensors and dynamic neural networks in Python with strong GPU acceleration
Home-page: https://pytorch.org/
Author: PyTorch Team
Author-email: packages@pytorch.org
License: BSD-3
Location: /usr/local/lib/python3.10/dist-packages
Requires: filelock, fsspec, Jinja2, networkx, nvidia-cublas-cu12, nvidia-cuda-cupti-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-runtime-cu12, nvidia-cudnn-cu12, nvidia-cufft-cu12, nvidia-
Required-by: fastai, torchaudio, torchdata, torchtext, torchvision

[ ] model_id = "dreamlike-art/dreamlike-diffusion-1.0"

pipe = StableDiffusionPipeline.from_pretrained(model_id, torch_dtype=torch.float16)
pipe = pipe.to("cuda")
```

```
./usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
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://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
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 models or datasets.
warnings.warn(
model_index.json: 100% |#####| 511/511 [00:00<00:00, 33.8kB/s]
Fetching 13 files: 100% |#####| 13/13 [00:22<00:00, 2.15s/file]
tokenizer/special_tokens_map.json: 100% |#####| 472/472 [00:00<00:00, 7.91kB/s]
text_encoder/config.json: 100% |#####| 612/612 [00:00<00:00, 10.7kB/s]
tokenizer/merges.txt: 100% |#####| 525k/525k [00:00<00:00, 3.75MB/s]
(./feature_extractor/preprocessor_config.json: 100% |#####| 518/518 [00:00<00:00, 4.35kB/s]
scheduler/scheduler_config.json: 100% |#####| 341/341 [00:00<00:00, 5.12kB/s]
model.safetensors: 100% |#####| 248M/248M [00:07<00:00, 42.8MB/s]
tokenizer/vocab.json: 100% |#####| 1.06M/1.06M [00:00<00:00, 4.90MB/s]
tokenizer/tokenizer_config.json: 100% |#####| 807/807 [00:00<00:00, 9.35kB/s]
unet/config.json: 100% |#####| 601/601 [00:00<00:00, 16.0kB/s]
vae/config.json: 100% |#####| 577/577 [00:00<00:00, 12.2kB/s]
diffusion_pytorch_model.safetensors: 100% |#####| 1.72G/1.72G [00:21<00:00, 166MB/s]
diffusion_pytorch_model.safetensors: 100% |#####| 167M/167M [00:06<00:00, 15.8MB/s]
Loading pipeline components...: 100% |#####| 5/5 [00:11<00:00, 3.49s/file]
```

```
def generate_image(pipe, prompt, params):
    img = pipe(prompt, **params).images

    num_images = len(img)
    if num_images>1:
        fig, ax = plt.subplots(nrows=1, ncols=num_images)
        for i in range(num_images):
            ax[i].imshow(img[i])
            ax[i].axis('off')
    else:
        fig = plt.figure()
        plt.imshow(img[0])
        plt.axis('off')
        plt.tight_layout()

[ ] prompt = "dreamlike, beautiful girl playing the festival of colors, draped in traditional Indian attire, throwing colors"
params = {}

[ ] generate_image(pipe, prompt, params)

100% |#####| 50/50 [00:12<00:00, 3.97s/img]
```

```
# negative_prompt
params = {'num_inference_steps': 100, 'num_images_per_prompt': 2, 'negative_prompt': 'ugly, distorted, low quality'}

generate_image(pipe, prompt, params)

100% |#####| 100/100 [00:50<00:00, 1.98s/img]

[ ] generate_image(pipe, prompt2, params)

100% |#####| 100/100 [00:50<00:00, 1.97s/img]
```

```
generate_image(pipe, prompt2, params)

100% |#####| 100/100 [00:50<00:00, 1.97s/img]
```



# RESULTS

VARIABLES LIKE THE QUALITY OF THE TRAINED MODEL, THE INTRICACY OF THE IMAGE DATASET, AND THE PARTICULAR PARAMETERS APPLIED DURING GENERATION CAN AFFECT THE OUTCOMES OF GENERATIVE AI'S CREATION OF REALISTIC IMAGES. GENERATIVE AI CAN PRODUCE STRIKINGLY LIFELIKE VISUALS, OPENING UP A WORLD OF POSSIBILITIES FOR ARTISTIC EXPRESSION, PRACTICAL APPLICATIONS ACROSS MULTIPLE INDUSTRIES, AND CREATIVE INQUIRY.

