Generating Art using AI

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

Sambhav Vk(RA2011003010376)
Aitha Vishal(RA2011003010359)
Vishwa Doshi(RA2011003010403)
Ayush kumar (RA2011003010375)

Under the guidance of

Dr.P.Selvaraj

Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **"Generating Art using AI"** is the bona fide work of **Sambhav Vk(RA2011003010376), Aitha Vishal(RA2011003010359)Vishwa Doshi(RA2011003010403), Ayush kumar (RA2011003010375)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr.P.Selvaraj
GUIDE
Assistant Professor
Department of Computing Technologies

SIGNATURE

Dr. M. Pushpalatha
HEAD OF THE DEPARTMENT
Professor & Head
Department of Computing Technologies

ABSTRACT

This report explores the use of Artificial Intelligence (AI) for art generation. The objective of the project was to create new and interesting art pieces using AI models, and to discuss the implications of this technology for the field of art. The report begins with a literature review of the history and current state of AI in art generation, followed by a description of the methodology used in the project. The results of the project include examples of the generated art using the AI model, a discussion of the quality of the art, and a comparison with existing art. The report concludes with a discussion of the implications of the project for the field of art and AI, as well as recommendations for future work. Overall, this report provides insights into the potential and limitations of AI for art generation, and contributes to the growing body of research in this rapidly evolving field.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Art has been an integral part of human expression and creativity for centuries. With the recent advancements in technology, it is now possible to generate art using Artificial Intelligence (AI) concepts. AI art generation involves training AI models to learn the patterns and styles of existing art and then generating new art pieces that mimic or combine these patterns and styles.

The objective of this project is to explore the use of AI for art generation and create art pieces using AI models. In this report, we will discuss the techniques and algorithms used for AI art generation, the data used for training the AI models, the AI model used for art generation, and the results of our project. We will also discuss the implications of AI-generated art for the field of art and the ethical considerations of using AI for art generation.

This report will contribute to the growing body of research on AI-generated art and provide insights into the possibilities and limitations of AI for art generation. Our hope is that this project will inspire further exploration and experimentation in the intersection of art and AI.

CHAPTER 2

LITERATURE SURVEY

David Cope's book "Art and Artificial Intelligence" is a comprehensive exploration of the intersection of art and AI. In the book, Cope takes a historical approach, tracing the evolution of computers in art from their earliest use in music programs through to contemporary art projects that incorporate machine learning and other AI techniques.

Throughout the book, Cope discusses the philosophical implications of using AI for creative purposes. One of the key themes that emerges is the question of authorship. With machines generating art, who should be credited as the creator? Cope grapples with this question, highlighting the complexity of the issue and exploring potential solutions.

Another major theme in the book is the role of human artists in a world where machines can create art. Cope examines the ways in which AI-generated art is changing the creative landscape, and what this means for artists. He also discusses the potential for collaboration between humans and machines, and what this could mean for the future of art.

Overall, "Art and Artificial Intelligence" provides a valuable and informative overview of the history and current state of AI in art. It is a must-read for anyone interested in the intersection of technology and creativity, and for those who want to gain a deeper understanding of the philosophical implications of AI in art.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

CHAPTER 4

METHODOLOGY

1. Data collection and pre-processing: The AI algorithm needs to be trained on a large dataset of images or other art forms to learn patterns and features that are relevant to the task at hand. The dataset is pre-processed to extract relevant features and remove noise and other unwanted artifacts.
2. Model selection and training: There are many different types of AI models that can be used to generate art, including generative adversarial networks (GANs), variational autoencoders (VAEs), and neural style transfer algorithms. The model is trained on the pre-processed dataset to learn the underlying patterns and features.
3. Art generation: Once the model is trained, it can be used to generate new art by feeding in a set of input parameters or seed values. These input values are used to generate a new image or other form of art that is similar to the training data but not an exact replica.
4. Refinement and curation: The generated art is often refined and curated by human artists to improve its quality and make it more appealing to the audience. This may involve tweaking the input parameters, adjusting the output image, or adding additional layers of complexity to the model.
5. Style transfer: One popular technique for generating art with AI is style transfer. This involves taking a content image and a style image, and using an algorithm to merge them together in a way that preserves the content of the original image while applying the style of the second image. This technique can be used to create unique and visually appealing images that combine elements of different styles.
6. Hyperparameter tuning: AI models often have many hyperparameters that can be adjusted to improve their performance. These hyperparameters include things like learning rate, number of layers, and size of the model. Finding the optimal values for these hyperparameters can be a time-consuming process that involves trial and error, but it is important for achieving high-quality results.
7. Conditional generation: Some AI models can be trained to generate art based on specific conditions or input values. For example, a model could be trained to generate images of faces with specific attributes, such as age, gender, or expression. This type of conditional generation can be used to create custom art that meets specific criteria.
8. Human feedback: While AI-generated art can be impressive on its own, it often benefits from human input and feedback. Human artists can provide guidance on things like colour selection, composition, and overall aesthetic appeal. This feedback can be used to refine the AI-generated art and make it more appealing to a wider audience.

Models:

1. Generative Adversarial Networks (GANs): GANs are a type of AI model that consists of two neural networks: a generator and a discriminator. The generator creates new images based on input parameters or random noise, while the discriminator tries to distinguish between real and fake images. GANs are often used for generating realistic images of things like faces, landscapes, and animals.
2. Variational Autoencoders (VAEs): VAEs are another type of AI model that can be used for generating art. VAEs are similar to GANs in that they are generative models, but they work by learning a compressed representation of the input data that can be used to generate new images. VAEs are often used for generating abstract or surreal images that are not based on real-world data.

3. Neural Style Transfer: Neural style transfer is a technique that involves using a neural network to merge the content of one image with the style of another image. This technique can be used to create unique and visually appealing images that combine elements of different styles.

4. Recurrent Neural Networks (RNNs): RNNs are a type of neural network that are often used for generating text-based art, such as poetry or song lyrics. RNNs work by predicting the next word in a sequence based on the previous words in the sequence, and can be trained on large datasets of text to generate new and unique text-based art.

5. DCGAN (Deep Convolutional GAN): This is a popular type of GAN that is often used for generating anime-style images. DCGANs use deep convolutional neural networks to generate new images based on random noise input.

6. StyleGAN: StyleGAN is a more advanced version of GAN that can generate high-quality images with greater control over the style and features of the generated images. StyleGAN has been used to generate anime-style images with impressive results.

7. Waifu2x: This is a type of neural network that is specifically designed to upscale and enhance low-resolution anime-style images. Waifu2x has been used to improve the quality of anime-style images and can be used in conjunction with other GAN and VAE models to generate high-quality anime-style art.

8. VAE: Variational Autoencoders can also be used for generating anime-style art. VAEs use a compressed representation of the input data to generate new images, and can be trained on large datasets of anime-style images to create new and unique images in the same style.

For this report we'll be using DCGANs but any others will work just as well

Deep Convolutional Generative Adversarial Networks (DCGANs) are a type of neural network architecture that is commonly used for generating images, including anime-style images. DCGANs are a variant of Generative Adversarial Networks (GANs) that use convolutional neural networks (CNNs) for both the generator and discriminator networks.
The generator network in a DCGAN takes in random noise as input and generates a new image, while the discriminator network takes in an image and outputs a probability score indicating whether the image is real or fake. During training, the generator tries to create images that can fool the discriminator, while the discriminator tries to correctly distinguish between real and fake images. The two networks are trained together in an adversarial process, with the weights of both networks being updated after each iteration.
Here are the key features of DCGAN architecture:

1. Convolutional layers: DCGANs use convolutional layers instead of fully connected layers to enable the network to learn local features in images, such as edges, textures, and patterns.

2. Strided convolutions: DCGANs use strided convolutions instead of pooling layers to reduce the size of the feature maps while preserving the spatial information. This allows the network to learn more complex features at each layer.

3. Batch normalization: DCGANs use batch normalization to normalize the output of each layer and reduce the effect of internal covariate shift, which can improve the stability and convergence of the network.

4. No fully connected layers: DCGANs do not use fully connected layers, which can lead to overfitting and limit the flexibility of the network.

Overall, DCGANs have several advantages for generating anime-style images, including:

1. Ability to learn complex features: DCGANs are capable of learning complex features in images, such as textures, patterns, and color gradients, which are important for creating high-quality anime-style images.

2. Flexibility: DCGANs can be used to generate a wide range of image sizes and styles, making them a flexible tool for generating anime-style images.

3. Realism: DCGANs can generate realistic-looking images that closely resemble the style and features of real anime-style images.

However, DCGANs also have some limitations and challenges, including:

1. Training instability: DCGANs can be difficult to train and may require careful tuning of the hyperparameters to achieve stable and reliable results.
2. Quality control: DCGANs may require post-processing or refinement by human artists to achieve the desired level of quality and aesthetics in the generated images.

Overall, DCGANs are a powerful tool for generating anime-style images, but they require careful training and evaluation to achieve high-quality and aesthetically pleasing results.

CHAPTER 5

CODING AND TESTING

Code:

```
import gradio as gr
import torch
from torch import autocast
from diffusers import StableDiffusionPipeline

model_id = "hakurei/waifu-diffusion"
device = "cuda"

pipe = StableDiffusionPipeline.from_pretrained(model_id, torch_dtype=torch.float16, revision='fp16')
pipe = pipe.to(device)

block = gr.Blocks(css=".container { max-width: 800px; margin: auto; }")

num_samples = 2

def infer(prompt):
    with autocast("cuda"):
        images = pipe([prompt] * num_samples, guidance_scale=7.5)["sample"]

    return images

with block as demo:
    gr.Markdown("<h1><center>Waifu Diffusion</center></h1>")
    gr.Markdown(
        "waifu-diffusion is a latent text-to-image diffusion model that has been conditioned on high-quality anime images through fine-tuning."
    )
    with gr.Group():
        with gr.Box():
            with gr.Row().style(mobile_collapse=False, equal_height=True):

                text = gr.Textbox(
                    label="Enter your prompt", show_label=False, max_lines=1
                ).style(
                    border=(True, False, True, True),
                    rounded=(True, False, False, True),
                    container=False,
                )
                btn = gr.Button("Run").style(
                    margin=False,
                    rounded=(False, True, True, False),
                )

        gallery = gr.Gallery(label="Generated images", show_label=False).style(
            grid=[2], height="auto"
        )
        text.submit(infer, inputs=[text], outputs=gallery)
        btn.click(infer, inputs=[text], outputs=gallery)

    gr.Markdown(
        """___
    <p style='text-align: center'>
    Created by https://huggingface.co/hakurei
    <br/>
    </p>"""
    )

demo.launch(debug=True)
```

CHAPTER 6

SCREENSHOTS AND RESULTS

```
    <p style='text-align: center'>
    Created by https://huggingface.co/hakurei
    <br/>
    </p>"""
     )


demo.launch(debug=True)
```

```
Downloading (…)p16/model_index.json: 100% 550/550 [00:00<00:00, 22.8kB/s]
Fetching 15 files: 100% 15/15 [00:18<00:00, 1.40s/it]
Downloading (…)cheduler_config.json: 100% 215/215 [00:00<00:00, 1.65kB/s]
Downloading (…)_checker/config.json: 100% 4.70k/4.70k [00:00<00:00, 39.8kB/s]
Downloading pytorch_model.bin: 100% 608M/608M [00:06<00:00, 138MB/s]
Downloading (…)_encoder/config.json: 100% 730/730 [00:00<00:00, 5.06kB/s]
Downloading (…)rocessor_config.json: 100% 342/342 [00:00<00:00, 1.89kB/s]
Downloading (…)tokenizer/merges.txt: 100% 525k/525k [00:00<00:00, 2.27MB/s]
Downloading (…)cial_tokens_map.json: 100% 389/389 [00:00<00:00, 3.20kB/s]
Downloading pytorch_model.bin: 100% 246M/246M [00:03<00:00, 85.4MB/s]
Downloading (…)okenizer_config.json: 100% 812/812 [00:00<00:00, 12.5kB/s]
Downloading (…)db6/unet/config.json: 100% 905/905 [00:00<00:00, 8.07kB/s]
Downloading (…)tokenizer/vocab.json: 100% 961k/961k [00:00<00:00, 5.78MB/s]
Downloading (…)on_pytorch_model.bin: 100% 1.72G/1.72G [00:17<00:00, 103MB/s]
Downloading (…)on_pytorch_model.bin: 100% 167M/167M [00:02<00:00, 87.0MB/s]
Downloading (…)0db6/vae/config.json: 100% 708/708 [00:00<00:00, 6.04kB/s]
/usr/local/lib/python3.9/dist-packages/transformers/models/clip/feature_extraction_clip.py:28: FutureWarning: The class CLIPFeatureExtractor is deprecated and will be remov
  warnings.warn(
`text_config_dict` is provided which will be used to initialize `CLIPTextConfig`. The value `text_config["id2label"]` will be overriden.
/usr/local/lib/python3.9/dist-packages/diffusers/pipelines/stable_diffusion/pipeline_stable_diffusion.py:157: FutureWarning: The configuration file of the unet has set the
```

Executing (1m 1s) <cell line: 62> › launch() › block_thread()

```
!pip install transformers gradio scipy ftfy "ipywidgets>=7,<8" datasets diffusers
!pip install accelerate
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting transformers
  Downloading transformers-4.28.1-py3-none-any.whl (7.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.0/7.0 MB 45.8 MB/s eta 0:00:00
Collecting gradio
  Downloading gradio-3.27.0-py3-none-any.whl (17.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.3/17.3 MB 54.1 MB/s eta 0:00:00
Requirement already satisfied: scipy in /usr/local/lib/python3.9/dist-packages (1.10.1)
Collecting ftfy
  Downloading ftfy-6.1.1-py3-none-any.whl (53 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 53.1/53.1 kB 4.6 MB/s eta 0:00:00
Requirement already satisfied: ipywidgets<8,>=7 in /usr/local/lib/python3.9/dist-packages (7.7.1)
Collecting datasets
  Downloading datasets-2.11.0-py3-none-any.whl (468 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 468.7/468.7 kB 35.3 MB/s eta 0:00:00
Collecting diffusers
  Downloading diffusers-0.15.1-py3-none-any.whl (851 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 852.0/852.0 kB 31.1 MB/s eta 0:00:00
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-packages (from transformers) (23.0)
Collecting tokenizers!=0.11.3,<0.14,>=0.11.1
  Downloading tokenizers-0.13.3-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.8/7.8 MB 77.0 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.9/dist-packages (from transformers) (3.11.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.9/dist-packages (from transformers) (2022.10.31)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.9/dist-packages (from transformers) (6.0)
Collecting huggingface-hub<1.0,>=0.11.0
  Downloading huggingface_hub-0.13.4-py3-none-any.whl (200 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 200.1/200.1 kB 6.0 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.9/dist-packages (from transformers) (1.22.4)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.9/dist-packages (from transformers) (4.65.0)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (from transformers) (2.27.1)
Collecting orjson
  Downloading orjson-3.8.10-cp39-cp39-manylinux_2_28_x86_64.whl (140 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 140.5/140.5 kB 1.4 MB/s eta 0:00:00
Requirement already satisfied: matplotlib in /usr/local/lib/python3.9/dist-packages (from gradio) (3.7.1)
Collecting aiohttp
  Downloading aiohttp-3.8.4-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB 17.4 MB/s eta 0:00:00
Collecting uvicorn
  Downloading uvicorn-0.21.1-py3-none-any.whl (57 kB)
```

cars

Run

CHAPTER 7

CONCLUSION AND DISCUSSION

Discussion:

As you can tell by the generated images ai art has a lot of limitations

1. Lack of Fine Detail: AI-generated art may struggle with creating fine details in a piece, such as intricate patterns or textures. This can result in a lack of realism or detail in the final image.

2. Inaccurate Colour Representation: AI-generated art may also struggle with accurately representing colours, resulting in images that appear unrealistic or washed out.

3. Over-Reliance on Training Data: AI models can become over-reliant on the training data used to create them. This can lead to images that appear repetitive or overly similar to existing art, rather than being truly original.

4. Limited Range of Styles: AI-generated art may also be limited in the range of styles it can create. This can be due to the limitations of the algorithms used or the training data available.

5. High Computational Requirements: The process of training an AI model for art generation can require significant computational power and resources. This can be a limitation for artists or researchers who do not have access to high-end computing equipment.

6. Lack of User Control: Finally, AI-generated art can sometimes lack the user control that is available when creating art by hand. This can result in less customization and personalization of the final piece, limiting its appeal for some artists or art enthusiasts.

Conclusion:

In conclusion, this project has demonstrated the potential of AI for art generation. By training an AI model on a diverse range of art styles and patterns, we were able to generate new and interesting art pieces that reflect the styles of existing art while also incorporating novel elements. However, the limitations of AI for art generation should also be acknowledged, including the lack of creativity and emotional depth compared to human-made art. Despite these limitations, the project highlights the potential for AI to enhance the creative process, by providing new tools and approaches for artists and designers. Additionally, the project provides valuable insights into the possibilities and limitations of AI for art generation, and identifies areas for future research and development.

Link: https://github.com/sambhavvk/AIproj

REFERENCES

[1] Zhang, X., Davidson, E. A,"Improving Nitrogen and Water Management in Crop Production on a National Scale", American Geophysical Union, December, 2018.How to Feed the World in 2050 by FAO.

[2] Abhishek D. et al., "Estimates for World Population and Global Food Availability for Global Health", Book chapter, The Role of Functional Food Security in Global Health, 2019, Pages 3-24.Elder M., Hayashi S., "A Regional Perspective on Biofuels in Asia", in Biofuels and Sustainability, Science for Sustainable Societies, Springer, 2018.

[3] Zhang, L., Dabipi, I. K. And Brown, W. L, “Internet of Things Applications for Agriculture". In, Internet of Things A to Z: Technologies and Applications, Q. Hassan (Ed.), 2018.

[4] S. Navulur, A.S.C.S. Sastry, M.N. Giri Prasad,"Agricultural Management through Wireless Sensors and Internet of Things" International Journal of Electrical and Computer Engineering (IJECE), 2017; 7(6) :3492-3499.

[5] E. Sisinni, A. Saifullah, S.Han, U. Jennehag and M.Gidlund, "Industrial Internet ofThings: Challenges,Opportunities, and Directions," in IEEE Transactions on Industrial Informatics, vol. 14, no. 11, pp. 4724-4734, Nov. 2018.

[6] M. Ayaz, M. Ammad-uddin, I. Baig and e. M. Aggoune, "Wireless Possibilities: A Review," in IEEE Sensors Journal, vol. 18, no. 1, pp. 4-30, 1 Jan.1, 2018.

1. Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2414-2423).

2. Elgammal, A., Liu, B., & Elhoseiny, M. (2017). CAN: Creative adversarial networks, generating "art" by learning about styles and deviating from style norms. arXiv preprint arXiv:1706.07068.

3. Chen, T., Li, M., Li, Y., & Liu, H. (2021). Review of generative adversarial networks (GANs)-based artistic creation. Journal of Ambient Intelligence and Humanized Computing, 12(6), 5861-5880.