Behind the Scenes of AI in Art

by

Rachel Molina

Submitted to the Department of Mathematics and Computer Science
School of SUNY Purchase College
in partial fulfillment of the requirements
for the degree of Bachelor of Arts

Purchase College
State University of New York

May 2023

Sponsor: Irina Shablinksy

Second Reader: Lee Tusman

Abstract:

The creative industry has been in turmoil since artificial intelligence (AI) has begun to generate its "own art." But AI was not always this intelligent, there are many past models that are milestones in AI's journey to image generation based on text prompts. Programs like OpenAI's DALL-E* are able to generate an image based of a user's text prompt, using a Contrastive Language-Image Pre-Training (CLIP) model, a diffusion model, and a prior model. Other programs created by smaller organizations or individuals, use some sort of image generator such as a generative adversarial network (GAN) plus CLIP to generate images based on text prompts. These AI programs have led to discussions regarding the legitimacy, or originality of these images, and the data images used to train the programs.

DALLE: The initial DALL-E released in 2021 did not use a prior model. The later version implemented the use of a prior model.*

Keywords: AI, Art, DALL-E, CLIP, GAN, Plagiarism, Deep Learning, Neural Networks, Neural Style Transfer, Python

Table of Contents

Chapter 1: Introduction

1.1: Research Question

21st century art has taken on a new medium-artificial intelligence (AI). In this paper, we aim to discover how these AI programs are able to generate images based on text prompts, while also taking a look at past models that have aided the development of current AI, and the plagiarism discussions that have emerged as a result of AI generating art.

1.2: Literature Review

The Integration of AI Technology into Art

We have all been witnesses to the tech-makeover of the 21st century. From projectors to smart boards, computers that span a room's width, and laptops, technology has become a necessity in our lives. Expanding alongside tech, artificial intelligence has also become a more integral part of our lifestyle, we see it in unlocking our phones, spam emails, search engines, maps, and even streaming platforms. Not only has artificial intelligence (AI) developed in the science field but it has also flourished in the arts.

We have seen AI in, so-to-speak, expected fields, so how has it worked its way into the arts? Within the last decade we have seen an increase in the digitalization of artworks, making it possible for someone in one part of the world to view a collection from a gallery across the globe. However, it is a challenging task to classify all existing art based on artist, artistic style, genre, medium, etc., and here is where AI has been lending a helpful hand. Earlier studies tackled the problem of automatically classifying artist, style, and genre, by extracting various handcrafted image features and employing different machine learning algorithms using those features (Cetinic & She). Application of convolutional neural networks (CNN) has furthered the classification accuracy. By supplying a large second database and pre-trained models, CNNs were showing further improvement in their original function as feature extractors, and in a variety of visual recognition tasks. Today's studies have diverged from classification and feature extraction to art creation (Cetinic & She). There have been many improvements in technology that have facilitated the interest in AI art, such as the development of rendering and texture synthesis algorithms in relation to computer graphics and computer vision. Although some of

these algorithms allowed users to modify images by applying a painterly or sketched style, they were still not generating new images.

There are many factors that contributed to the emergence of AI art, including but not limited to, the use of deep neural networks, and neural style transfer (NST). Neural style transfer was popularized by Gatys, who used CNNs to create stylized images by separating and combining the image "content" and "style", therefore automating photo manipulation (Cetinic & She). GAN, introduced by Goodfellow, used two trained competing models, a generator, and a discriminator, that are typically implemented as neural networks (Goodfellow). The generator's job is to "capture the distribution of true examples of the input sample and generate realistic images", and the discriminator's job is to classify generated images as fake and the real images from the sample as actually real. GAN's optimization was designed as a minimax, that is, optimization was reached at a saddle point; A minimum in relation to the generator and a maximum in relation to the discriminator (Cetinic & She). Elgammal decided to take it one step further and introduced "an AI creative adversarial network", AICAN (Cetinic & She). He argued that if the GAN model was trained on images of paintings, then it would begin to generate images of already existing paintings. This method was still not producing anything new, so they proposed changing the optimization criteria so that the network would be able to create art by "maximizing deviation from established styles while staying within the art distribution" (Cetinic & She). This proved to be successful and in many experiments the authors of AICAN found that the audience was often not able to distinguish the AI generated art from art produced by humans.

In recent years there has been an increase in transformer-based architecture, specifically in their application of text. In January 2021, OpenAI introduced *DALL-E,* a 12 billion parameter

network trained to generate images based off text description, using a dataset of text-image pairs (Cetinic & She). *DALL-E* is an AI program that creates art based on a user's text description. These kinds of programs have been improved and advanced with the recent surge of interest in cross-modal models, which is any kind of learning that involves information obtained from more than one modality. Radford introduced the *CLIP (*Contrastive Language-Images Pre-Training) model that was trained on 400 million image-text pairs collected from the internet (Radford). The *CLIP* model jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. Many programs, including *DALL-E**, BigSleep, and DeepDaze, rely on the *CLIP* model to rank the results of generated images in relation to the input text.

Generating art using artificial intelligence is a relatively new idea and has led to many discussions. Some of those discussions revolve around fundamental questions, where AI fits into the history of visual arts, whether AI is replacing human creativity, questions about copyright claims, along with many other important questions. AI technology has also led to the second industrialization of art production, shifting from mechanical reproduction to digital reproduction of art (Tao). Unlike photography, AI has not directly created any new art forms, but instead is on its way to replacing artists on the artistic production line (Tao).

The technological advancements of the 21st century have allowed for AI technology to become more advanced and have aided the adoption of AI into art. As technology has improved throughout the years AI has transitioned from being an art classification tool to becoming its own artist and creating new images. But its transition has not been welcomed by all, many discussions have arisen regarding the fundamentals of art, copyright, and whether or not AI art can be

classified as “art”. Tao, Cetinic and She have all done an excellent job exploring these issues and explaining the history and development of AI art.

Chapter 2: Artificial Intelligence

2.1: Introduction/Overview

In recent years, artificial intelligence (AI) has become an extremely popular and fast-growing field. Many tasks that once required a great deal of time have become automated through artificial intelligence and machine learning. Astonishingly, we interact with AI almost every day, from voice assistants to image recognition for face unlock in cell phones. AI has integrated itself into our style of living.

Artificial intelligence is defined by Britannica as the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings, or humans, because they require human intelligence and discernment (Copeland). Currently, there is no single machine or AI that can perform a wide variety of tasks that an ordinary human can do, but there are some AIs that can match or even outperform humans in specific tasks.

The birth of artificial intelligence occurred in 1950 in a paper written by Alan Turing (“Computing Machinery and Intelligence”), in which he proposes the question, “Can machines think?” Turing also proposed a test, now famously known as the “Turing Test”, where a human interrogator would try to identify the difference between a computer and human text response. The original Turing test required 3 terminals, each of which was physically separate from the other two. One terminal is operated by a computer, while the other two are operated by humans. During the test, one human is the questioner, while the second human and the computer act as the respondents. After a certain amount of time or a number of questions, the questioner may attempt to discern which respondent is human and which is the computer. This test is repeated

many times, and if the questioner makes the correct determination in half of the test runs or fewer, the computer is considered to have artificial intelligence. This is because the questioner regards it as "just as human" as the human respondent (George and Gillis).

AI can be divided into 2 categories, a handcrafted knowledge or rule-based approach, and a machine learning approach. AI programs in the handcrafted knowledge approach are older and typically consist of programmed rules. Rule-based programs try to represent human knowledge into programmed sets of rules that computers can use to process information. The "intelligence" of handcrafted knowledge systems is simply a very long list of rules in the form of "if given x input, then provide y output" (Allen). Unlike handcrafted knowledge systems, machine learning programs generate their own rules. Machine learning refers to the study of computer systems that learn and adapt automatically from experience, without being explicitly programmed to do so (Coursera). To generate these rules, humans provide the system with training data. By running a human-generated algorithm on the training dataset, the machine learning system generates the rules such that it can receive input *x* and provide the correct output *y* (Allen). Although machine learning systems are responsible for the recent advancements in AI technology, there are still areas in which rule-based approaches perform better than machine learning approaches. One of these areas is tax preparation software. By requiring users to input their tax information according to pre-specified data formats and then processing that data according to the formally programmed rules of the tax code, the output can be good enough to pass an IRS audit (Allen). On the other hand, there are many areas such as chess playing, language translation, and image classification, in which machine learning programs have outperformed knowledge-based systems (Allen).

2.2: Machine Learning & Deep Learning

Machine learning also has many sub-categories such as, supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. Supervised learning means that before the algorithm processes the training data, some "supervisor", which may be a human, a group of humans or a different software system, has accurately labeled each of the data inputs with its correct associated output. For example, in a proposed image classification system, the goal is to classify the object in the image as either "cat" or "dog", the labeled training data would have image examples paired with the correct classification label (Allen). Once the algorithm is finished training, it no longer requires pre-trained data. Unsupervised learning algorithms are those that use data but do not require labels for the data. Typically, these systems have lower performance than supervised learning, but these systems can be used in places where supervised learning models are not viable. Semi-supervised learning uses both labeled and unlabeled data and has a mix of pros and cons of supervised and unsupervised learning (Allen). Lastly reinforcement learning is a training method based on rewarding desired behaviors and/or punishing undesired ones. In general, a reinforcement learning agent (which is an agent trained to complete a task within an uncertain environment) is able to perceive and interpret its environment, take actions and learn through trial and error (Carew). An example of machine learning is IBM (International Business Machines)'s Watson. In February 2011, IBM's Watson computer competed on *Jeopardy!* against two of the show's biggest all-time champions and won. Watson is a computer running software called DeepQA, developed by IBM research. IBM's scientists have said that Watson does not actually think, and David Ferrucci who spent 15 years working at IBM research said "'The goal is not to model the human brain'" (IBM). Watson is a

good illustration of machine learning techniques, because unlike deep learning techniques machine learning does not aim to replicate the human thought process.

Deep learning is another kind of AI, which is a machine learning technique that layers algorithms and computing units-or neurons-into artificial neural networks that mimic the human brain (Coursera). Deep learning can be thought of as an evolution of machine learning. Unlike machine learning which may require human intervention when the output is wrong, deep learning algorithms can improve their outcomes through repetition without human intervention. These repetitions require a large, and sometimes unstructured data set, contrasting the relatively small set of data that machine learning requires (Coursera). Deep learning techniques can be applied to any of the aforementioned machine learning sub-categories. Deep learning neural networks, or artificial neural networks, attempt to mimic the human brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data (IBM). A deep neural network has multiple layers of interconnected nodes (a point in a network where data or communication can enter or leave (A.I For Anyone)). The first layer is the input layer, which is where the model gets the data that will be processed. The output layer is also visible, and this is where the final prediction or classification is made. Each inner layer builds upon the previous layers, refining or optimizing the prediction or categorization, this is known as forward propagation (IBM). To improve the model's accuracy, there is a process used called backpropagation. Backpropagation uses algorithms such as gradient descent, to calculate the errors in predictions and then adjust the weights and biases of the function by moving backwards through the layers to train the model (IBM). Forward propagation and backwards propagation work together to improve the model's accuracy overtime. This is just one oversimplified example of a deep learning model, as there are

many much more complicated models such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs). Convolutional neural networks are often used in computer vison to assist in interpreting images as 2D arrays (3D if they are color pictures). One application of CNN's is facial recognition. After the image has been received as an input, the network outputs a set of values that indicate the attribute of the faces or facial features at various point in the image. Then those outputs are compared to existing data to match a face with a name (Gandharv). This is just one application of CNN's and deep learning models.

Natural Language Processing

Natural language processing (NLP) is a branch of Artificial Intelligence, that focuses on trying to give computers the ability to understand text as humans do (IBM Cloud). NLP combines linguistics, machine learning, and deep learning models, to allow computers to process human language in the form or text or voice data. Applications such as customer service bots, voice operated global positioning system (GPS), autocorrect, digital assistants, and plagiarism checkers all use NLP.

Natural language processing is needed to help computers understand the information or text they are given. NLP also resolves ambiguity and provides important syntactic and semantic understanding. Syntactic understanding is the ability to understand the structure of sentences and the relationships between the words within them. Semantic understanding refers to the ability to understand meaning of the language. Natural language is difficult to understand for a variety of reasons including nonstandard English, segmentation issues, idioms, neologism, word knowledge, and entity names.

The first step in natural language processing is called tokenization. Tokenization refers to breaking strings into tokens, which are small structures or units. Next either or both, stemming

and lemmatization occur. Stemming refers to normalizing words into its stem or root form. This is done by removing the end or beginnings of a word, considering a list of common prefixes, and suffixes. For example, the words affection, affects, and affected all originate from the root word affect. The stem of a word is a rough approximation of the base form, which may or may not be a valid word in the language. This can lead to errors, which is why, although it is computationally less expensive, it is often considered a second choice to lemmatization. Lemmatization reduces words to their base or dictionary form, which is known as the lemma. For example, the lemma of the word “cats” is “cat”, and “went” is “go”. Lemmatization takes into consideration the morphological analysis of the word and requires a vocabulary or detailed dictionary. Then each token is assigned its part of speech (POS) tag, this process is referred to as POS tagging. Next named entities in the document are classified and identified, this is known as named entity recognition (NER). Entities include people, places, organizations, and dates. Lastly related words are grouped together. This is called chunking; the words are grouped together based on their part of speech and syntactic structure (MonkeyLearn).

The natural language processing field has drastically improved in the last couple years. One of the key breakthroughs in NLP has been the development of transformer-based models such as BERT, GPT-2, and GPT-3. What makes these models so great, is the self-attention mechanism, which allows them to learn contextual relationships between words and phrases in text, enabling more accurate generation and natural language output. Most recent was the development of chatbot, ChatGPT by OpenAI, which is built on top of their previous models, GPT-3, and GPT-4. ChatGPT uses a neural network architecture and unsupervised learning to generate responses (Majumder).

The development of transformer-based models and the self-attention mechanism has greatly aided the NLP field. These advancements have led to increased accuracy and more natural language outputs in a variety of NLP tasks. As technology improves and research continues, we can only expect NLP to continue to advance and potentially more sophisticated and intelligent language-based models to be developed.

AI has rapidly advanced in recent years and has become an essential part of our daily lives. Artificial intelligence can be divided into two categories, handcrafted or rule-based approaches, and machine learning approaches. Although machine learning programs are more advanced than rule-based systems in areas such as language translation, image classification and chess playing, rule-based systems perform better than machine learning approaches in areas such as tax preparation. There are different sub-categories in machine learning such as supervised, unsupervised, semi-supervised, and reinforcement learning. One of these sub-categories is deep learning, which has evolved to mimic the human brain and can improve its outcomes without human intervention. As technology and AI continue to advance it will be exciting to see how various industries are affected and how AI become more and more integrated into our daily lives.

Chapter 3: AI in Art

3.1: Introduction/Overview

When artificial intelligence is referred to in art, it often means one of two things, either AI is begin used in art classification or AI is begin used to generate art. AI can be used to assist museums or art galleries in processing, digitizing or classifying art by labels such as artist, style, or genre (Cetinic and She). On the other hand, recently, AI has been used to generate images based off user's text prompts. Companies like OpenAI have developed models such as DALL-

E*, which use a CLIP model, a diffusion model and a prior model to generate images based on users' text prompts.

Although today's models are remarkable, there are almost a decade's worth of past models that have aided the development of current models. In 2014 Ian Goodfellow, a PhD student at the University of Montreal, and his colleagues published a paper titled "Generative Adversarial Networks" (History of Data Science). Goodfellow's paper proposed a model that contained two sub-models, a generator and a discriminator. The generator would generate images or samples to try and fool the discriminator, while the discriminator would try to distinguish the true data from the generator samples. Today many art or image generative AI programs use some variety of a GAN model in their models.

The following year, Mordvintsev, a Google engineer, introduced DeepDream-a computer vision algorithm that uses convolutional neural networks (CNNs) to find and enhance patterns in images (Mordvintsev, Olah and Tyka). As TensorFlow says, it is similar to when a child watches clouds and tries to interpret random shapes, DeepDream over-interprets and enhances the patterns it sees in an image.

Later in 2016, neural style transfer (NST) was introduced by Gatys, Ecker and Bethge. The NST technique builds upon the DeepDream algorithm, but instead of amplifying detected patterns in the image, the NST technique transfers the style of one image onto the content of another image.

Artificial Intelligence Creative Adversarial Network (AICAN) was introduced in 2017 by Dr. Ahmed Elgammal and is the model closest to the art generative models we have today. AICAN was able to generate original artwork using CANs. The system is trained on existing art, which it uses to generate new art that is unique but may share some characteristic with the art

from the training data set. The creative adversarial network (CAN) differs from GAN because in the CAN, the generator receives two signals from the discriminator for any work it generates (Elgammal, Liu and Elhoseiny). What these signals are and why the generator receives two, will be discussed in more detail in the following section.

Most recently, in 2021, was the introduction of DALL-E by OpenAI. The following year OpenAI released DALL-E 2, which combines a CLIP model, a diffusion model and a prior model to generate new images based on text prompts. CLIP was also created by OpenAI and was first introduced in a paper titled "Learning Transferable Visual Models from Natural Language Supervision". Unlike any of the previous models mentioned, DALL-E was the first model to generate images based on text.

The integration of AI and art has come a long way since the introduction of generative adversarial networks in 2014. Artificial intelligence has gone from a museum assistant who digitized and classified art, to what some consider an artist, creating new work based on user prompts. In the following section, we will dive into the technical details of some of these models and explore how they work.

DALLE: The initial DALL-E released in 2021 did not use a prior model. The later version implemented the use of a prior model.*

3.2: Art generative AI model Milestones

The following models are considered milestones in the advancement of today current AI models.

GAN 2014

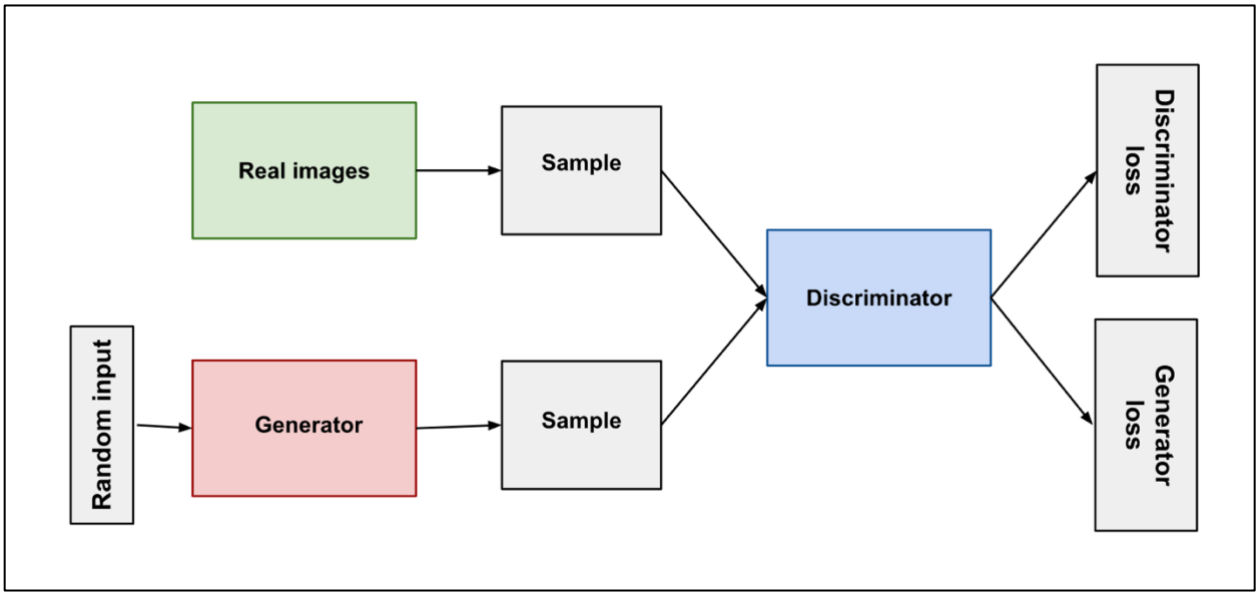

Figure 1: Overview of GAN Structure (Google Developers)

A GAN model contains two sub models called the generator and discriminator. Both sub models are trained at the same time through a process called adversarial training. During the training, the generator and discriminator are trained in alternating steps. The generator is trained to create new synthetic data that is similar to the real data from the training set. The discriminator is trained to distinguish the real data samples and the synthetic ones created by the generator. The optimization is designed as a minimax problem, meaning optimization has been reached at a saddle point, which is a minimum in relation to the generator and a maximum in relation to the discriminator (Cetinic and She).

The Discriminator

The discriminator connects two loss functions (seen in figure 1), and during training it ignores the generator loss, focusing only on the discriminator loss. During training the discriminator classifies both real data and fake data from the generator. The discriminator loss penalizes the discriminator for misclassifying a real instance as a fake or a fake instance as real

(GoogleDevelopers). Lastly through backpropagation from the discriminator loss through the discriminator network, the weights are updated.

The Generator

The generator is a generative model, meaning it tries model to how data is placed throughout the space. An example of a generative model, by GoogleDevelopers, is as follows:
There are IQ scores for one thousand people, and the distribution is modeled by the following procedure:

1. Roll three six-sided dice.
2. Multiply the roll by a constant *w*.
3. Repeat 100 times and take the average of all the results.

Try different values for *w* until the procedure results equal the average of real IQ scores.
In this scenario, every roll is essentially generating the IQ of an imaginary person.

The generator learns to create fake data by incorporating feedback from the discriminator-It learns to fool the discriminator into classifying its output as real (Google Developers). Typically, the generator is implemented as a deep neural network where the input is random noise. Then a differential function is used to map the input noise into the data space, while also minimizing the log probability of the discriminator being able to correctly classify its synthetic data samples as not coming from the real data distribution (Goodfellow, Pouget-Abadie and Mirza).

DeepDream 2015

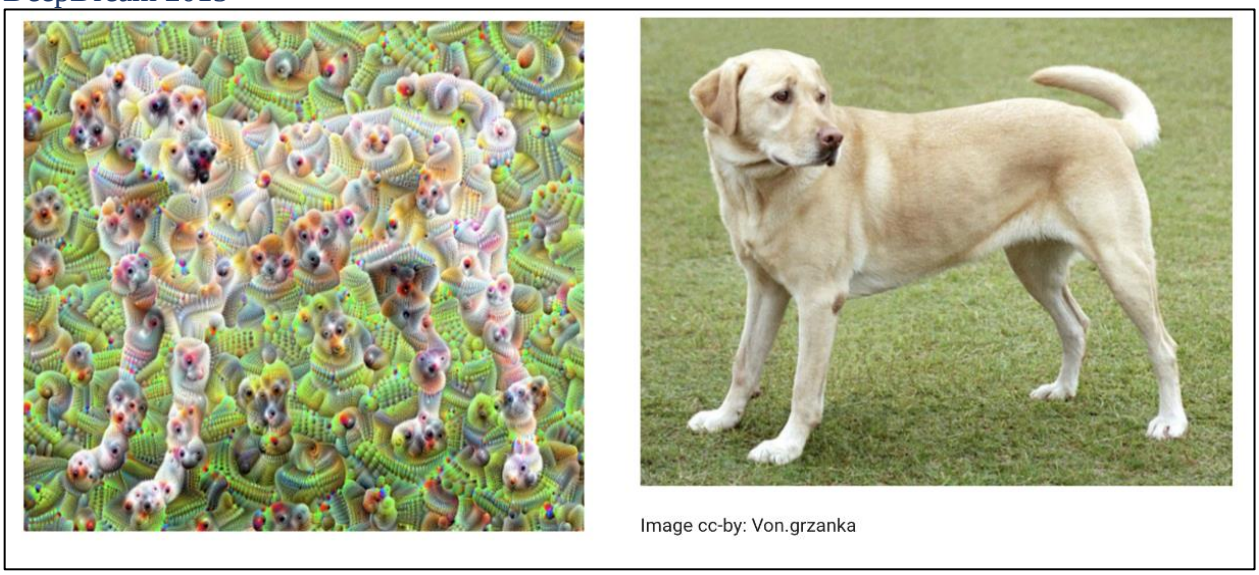

Figure 2: (On the right) The input image of a dog. (On the left) The resulting image after DeepDream (Abadi, Agarwal and Barham)

The DeepDream AI uses neural networks, typically a pretrained convolutional neural network (CNN), trained on millions of images, to recognize patterns in an image and over enhance the pattern. By showing the networks images of what was desired, it will eventually extract the essence of the matter. Once those features have been acquired, the AI can be asked to enhance those layers. CNNs generate feature maps, which are over-enhanced in DeepDream to generate “dream-like” images (TensorFlow). The artificial neural network is trained on millions of training examples and by gradually adjusting the network parameters (back propagation) until it gives the wanted classification. After training, each layer progressively extracts higher and higher-level features of the image, until the final layer essentially makes a decision on what the image shows. Perhaps the first layer looks for edges or corners, the intermediate layers interpret the basic features to look for overall shapes or components, like a door or a leaf. The final few layers assemble those into complete interpretations-the neurons here activate in response to complex things such as entire buildings or trees (Mordvintsev, Olah and Tyka).

NST 2016

Figure 3: (Left) The input image. (Middle) The style desired to apply onto the input image. (Right)The resulting image (Abadi, Agarwal and Barham).

NST is an optimization technique used to take two images, one for content the second for style, and combines them so that the image produced looks like the content image but "painted" in the style of the style image. The end image is obtained by optimizing the output image to match the content statistic of the content image and the style statistic of the style reference image (TensorFlow). These statistics are extracted from the images using a convolutional network. The first layers of the network represent the low-level features, and the last few layers represent higher-level features-object parts like eyes, or wheels. The representations of the content and style from the images, are defined in the intermediate layers (TensorFlow). These layers are able to define style and content representations because in order for the network to perform image classification, it needs to understand the image. Meaning when the raw image is taken in as input pixels the network must build an internal representation that converts these pixels into a complex understanding of the features present in the image. During the optimization process the intermediate layers are extracted from the content image and from the style image. Then the intermediate layers extracted from the style image are applied to a copy of the content image, or a random image. Lastly the pixel values are of the new generated image are adjusted to minimize the difference between the style features of the style image and the content features of the content image.

AICAN 2017

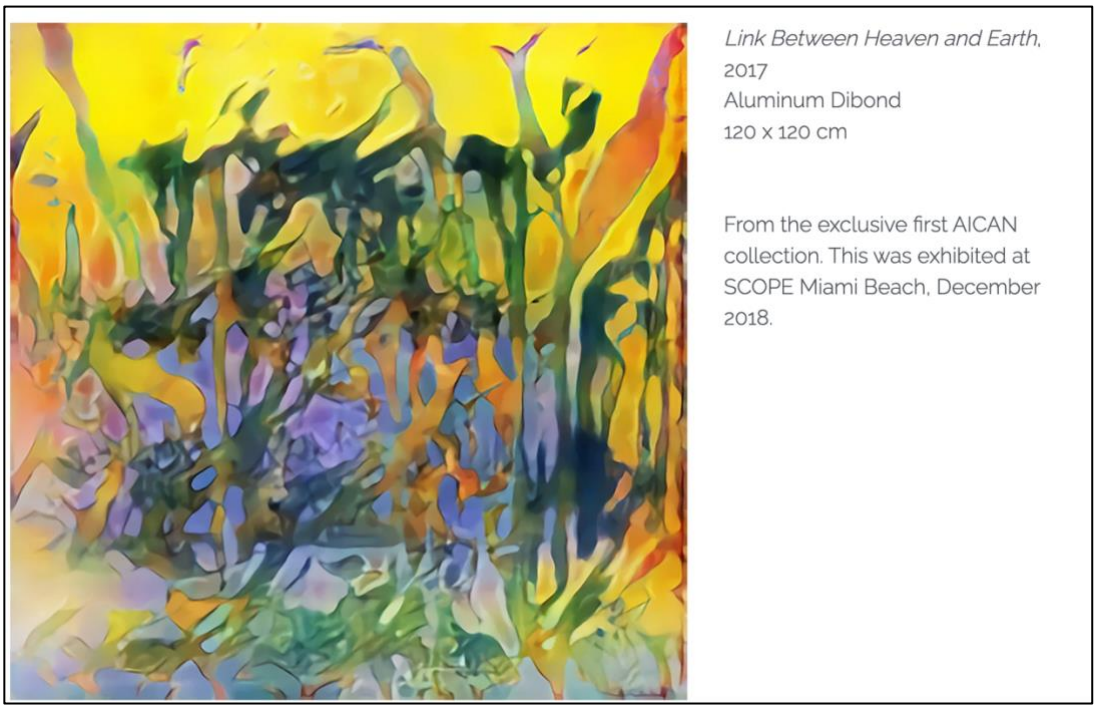

Figure 4: An image produced using AICAN titled: Link Between Heaven and Earth (AICAN)

AICAN relies on a CAN model, meaning just like GAN, it contains two sub models, a generator, and a discriminator. In a basic CAN model the discriminator has the same task as a discriminator in a GAN model, however in AICAN the discriminator assesses the generated image on multiple criteria, including their realism, style, quality and creativity.

The Discriminator

The discriminator has access to a large set of art associated with style labels, while the generator does not. Using that set, it learns to discriminate between art styles. The discriminator in the CAN model differs from the discriminator in the GAN model, because here discriminator also evaluates the creativity and originality of the generated art in addition to distinguishing between real and generated art. Its evaluation is based on the visual similarity between the generated images and the real images in the dataset, as well as the diversity, originality, and creativity of the generated images. During training, the discriminator distinguishes images and

assigns a score or probability to each image that reflects how likely the images is to be real. The term “real” refers to how closely the generated image resembles real-world images, especially those from the dataset (Elgammal, Liu and Elhoseiny). It then generates a second score that measures the image’s creativity and diversity.

The Generator

The generator generates art starting from random input, but unlike GAN, it receives 2 signals from the discriminator. The first signal is the classification of “art or not art”, or the first probability score. This signals whether the discriminator thinks the generated art is coming from the same distribution as the actual art it knows about. In a traditional GAN, this signal enables the generator to change its weights to generate images that will more frequently deceive the discriminator as to whether the image is coming from the same distribution. The second signal is about how different the generated art is from the real images in the training dataset, which encourages the generator to produce more original art differing from the original images (Elgammal, Liu and Elhoseiny). The originality or exploration of new styles and adherence to established styles can be controlled by the weighting of the diversity-promoting loss function. This function penalizes the generator for producing images that are too similar to each other and rewards it for producing images that are diverse and unique (Elgammal, Liu and Elhoseiny).

DALL-E 2021 & 2022

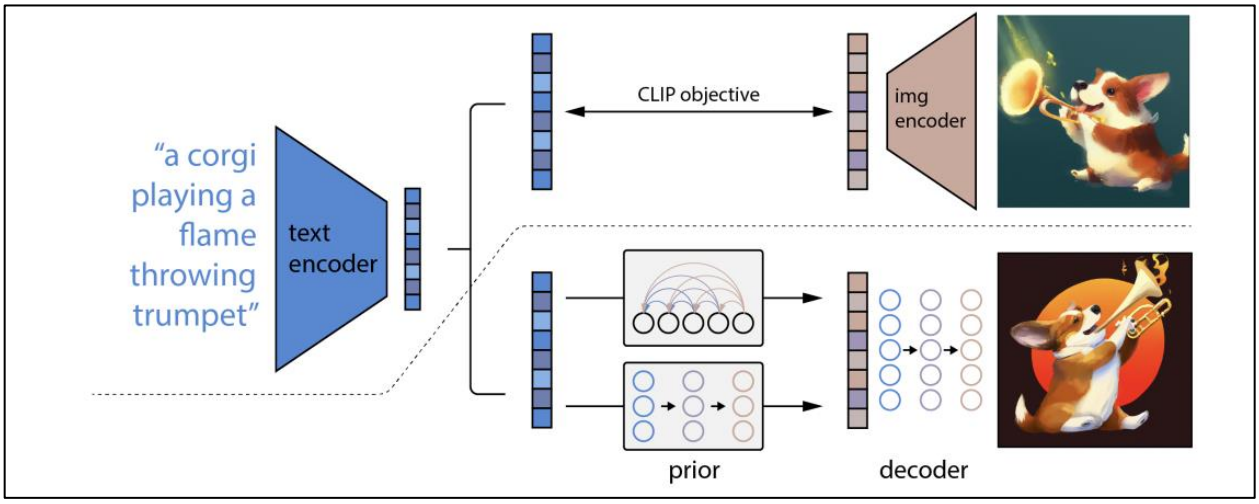

Figure 5: Above the dotted line, is the CLIP training process, in which a joint representation space for text and images is learnt. Below the dotted line, is the text-to-image process. The CLIP text encoding is first fed to an autoregressive or diffusion prior to produce an image encoding, then this encoding is used to condition a diffusion decoder which produces the final image (Ramesh, Dhariwal and Nichol).

DALL-E is a decoder-only transformer that receives both the image and text as a single stream of 1280 tokens-256 for text and 1024 for the image-and models them all autoregressively (Wang). The first version of DALL-E released in January 2021, and did not use a CLIP model. The updated version of DALL-E was released the following year in April and unlike the first version it did use CLIP. For the rest of the paper when we refer to DALL-E, we will be talking about the updated version. DALL-E relies heavily on CLIP, another model also made by OpenAI. CLIP is a pre-trained model, that is able to understand the relationship between images and text, allowing it to perform many tasks such as image classification, object detection, and image captioning. DALL-E first uses CLIP to turn the textual descriptions into text encodings (ignoring the resulting image encodings), those encodings are the input for a prior model-called a diffusion prior. Knowing CLIP creates image and text encodings, one might ask why the prior is needed. OpenAI ran various experiments that proved the results using a prior model were vastly superior to those without (see figure 6). The diffusion model then outputs an image embedding

which is used to condition a diffusion decoder which produces the final image. During the prior and decoder training, the CLIP model is frozen (Ramesh, Dhariwal and Nichol).

Diffusion Prior

Diffusion models are transformer based generative models. The models take in some input, for example a photo, and add noise over timesteps, until the photo is unrecognizable. Then the model tries to rebuild the image to its original form. Through this process the model learns to generate the data, or in this case images.

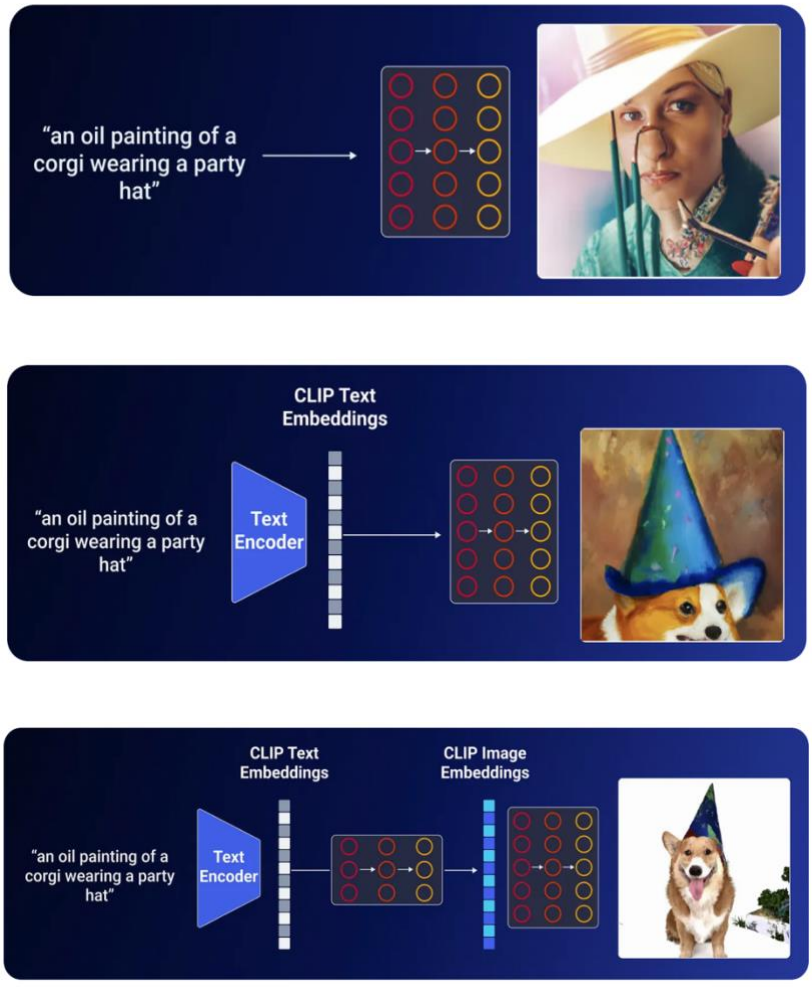

Figure 6: Aditya Singh highlighted the difference talked about by OpenAI. The first image is generated by passing the caption directly to the decoder. The second image is generated by passing the CLIP encoding to the decoder. The final image is generated by giving the CLIP text encoding to the prior and then giving the resulting prior image encoding to the decoder (Singh).

The models discussed have played enormous roles in development of the abilities of AI today. As these models continue to advance, more models are developed and technology improves, artificial intelligence will only get better.

Chapter 4: Trying to Re-Create Art Generative AI

4.1: The Plan

Originally, I planned to try to re-create one of the models I learned about or to re-create a different one that still generated art based on text prompts. After learning about all the models, I realized that most of the newer models used some kind of image generator in combination with CLIP, and that I was being overambitious. I did not have the experience or knowledge to be able to create my own art generative AI. Most of the programs we see today are created by teams of people who have years of experience in this field, or people who just have experience. Despite the vast research I had done, I still did not have the expertise to be able to code a model from scratch. So, my new goal was to learn enough that I would be able to at least propose a kind of art generative (based on text prompts) program that would use other pre-existing models.

4.2: Observations

4.2A: Python & IDEs

```
In [19]: #exercise 2, user calculator
         first_number = input("First number: ")
         second_number = input("Second number: ")

         total = float(first_number) + float(second_number)
         print("Sum "+ str(total))

         First number: 10
         Second number: 250
         Sum 260.0

In [26]: #string manipulation/ learning about objects and methods
         course = "Learning Python"
         print(course.upper())
         print(course.lower())
         #the .upper() method makes all characters in the string capitals, the .lower() makes them lower case

         course.find('n')
         #.find() find the first occurence of what you are "searching" for, and returns the index(this is lo

         course.find('Python')
         #returns the index of the word
         #can also use:
         # print("Python" in course)
         course.replace('Python', 'Java')
         #(word to be replaced, its replacement)

         LEARNING PYTHON
         learning python
         True
Out[26]: 'Learning Java'
```

Figure 7: A small portion of the practice code I had written while following along the videos. The whole file can be found at this link: https://github.com/rmolinazea/container.git

I very quickly realized that to both construct or re-create one the models and to understand the code of the models, I would need to learn Python. Python is the number one language for AI development, which led to a problem because I did not know anything about Python. This led to another problem, I did not know what integrated development environment (IDE) to use. I have the most experience with Java, and up until then I had done all my programming in IDEs like Replit, IntelliJ, or P5.js, so I had no idea what IDE I was supposed to use. Upon consultation with my mentor, Dr. Irina Shablinksy, I was led to the IDE Anaconda, which had Jupyter Notebook. I was also unfamiliar with this IDE, so I watch a couple videos and read a few articles that helped me understand the basic so that I could navigate the environment. Next, I had to learn Python. So, I followed along two videos: *Python for Beginners-Learn Python in 1 Hour* by Programming with Mosh and *Python Tutorial-Python Full Course for Beginners* also by Programming with Mosh. These videos gave me a basic understanding of the language,

and I followed along in the Jupyter Notebook (see figure 7). While the videos and numerous articles I went through were informative I cannot say I am anywhere near an expert in Python. Nonetheless, the effort I put in enabled me to develop some familiarity with the code used in the models I researched.

4.2B: Neural Networks

```
In [2]:
import numpy as np
#assign the input values
input_value = np.array([[0,0],[0,1],[1,1],[1,0]])
input_value.shape
input_value

Out[2]:
array([[0, 0],
       [0, 1],
       [1, 1],
       [1, 0]])

In [ ]:
#assign the output values

In [3]:
output = np.array([0,1,1,0])
output = output.reshape(4,1)
output.shape

Out[3]:
(4, 1)

In [ ]:
#assign our weights

In [4]:
weights = np.array([[0.1],[0.2]])
weights

Out[4]:
array([[0.1],
       [0.2]])

In [ ]:
#now the bias(which is a constant)

In [5]:
bias = 0.3
```

Figure 8: A snippet of the code from my experience learning about neural networks, the full notebook can be found at this link: https://github.com/rmolinazea/container

The basis of most deep learning models are neural networks, and although much can be learned by reading there is another layer of experience gained by coding firsthand. I followed along a video by edureka! on YouTube titled *Neural Network Python | How to make a Neural Network in Python | Python Tutorial | Edureka*, the video not only explained how neural networks work, but also included instructions on how to code the neural network in the video. The neural network in the video took in an array with 2 elements of either 0 or 1 and outputted a 0 or 1. To begin, we gave the network the training data, the expected outputs, initial weights, and

bias. This network uses the sigmoid function as the activation function and requires the derivate of the function as well as the learning rate.

4.2C: Natural Language Processing (NLP)

I initially thought that most of these art generative programs had to have some NLP program involved to handle the text prompt, which led to a lot of research regarding the matter. One program I learned about was Valence Aware Dictionary and sEntiment Reasoner (VADER). VADER is a rule-based sentiment analyzer, which contains a list of lexical features that are generally labeled as per their semantic orientation as positive or negative (Hutto and Gilbert). At the time, DALL-E was the main focus of my research and so the next subject I researched was how DALL-E handled the text prompts. What I found, was that DALL-E relied on OpenAI's CLIP for both text processing and image processing.

4.3: The Code

4.3A: Google Colab

I realized very quickly that these programs would require a lot of computing power, meaning my laptop would not cut it. I could use the school computers but I am a commuter student so that would not work long term. It was when researching about StableDiffusion that I came across Google Colab. Google Colab is a "hosted Jupyter notebook service" that allows anyone to write and execute python code through the browser and provides free of charge access to computing resources including GPUs. (Google). So, although I had learned how to use Jupyter Notebook, most of the code I would be looking at would be better suited to the Google Colab environment since I could change the runtime type to use the GPU.

4.3B: CLIP

CLIP is one of the main elements of text-to-image programs, such as DALL-E, StableDiffusion, and Midjourney. I knew that I would most likely be using it in whatever program I ended up proposing, so I needed to have a good understanding of how the program

worked. My first mistake was assuming that because I understood how CLIP worked, I would understand the code. The biggest issue was all the code I was reviewing was in Python, and although I had become familiar with Python, I was far from being able to understand why certain libraires were imported or what they did. So, before I could analyze what was happening in the CLIP code, I needed to go through step by step and understand why certain libraries were imported, which involved a lot more researching.

1) TorchVision: One reason this library was imported, was because it handles the image preprocessing. CLIP uses the Transform method, to resize, center crop, convert to RGB, and normalize the images. Rather than doing all of these tasks individually, CLIP uses the Compose method which allows multiple transformations to be turned into one big transformation.
2) OS Package: This package is used to navigate the operating system-dependent functionality.
3) Skimage: Skimage is an open-source library of algorithms for image preprocessing. CLIP uses Skimage for their images and text descriptions. The notebook in which I documented my process of learning skimage can be found here: https://github.com/rmolinazea/container.git

After I learned what the libraries were for, I went through the CLIP notebook code and tried entering my own examples to get first-hand experiences rather than solely using the notebook's examples, my edited version of the CLIP notebook can be found here: https://github.com/rmolinazea/container.git.

4.3C: VQGAN+CLIP

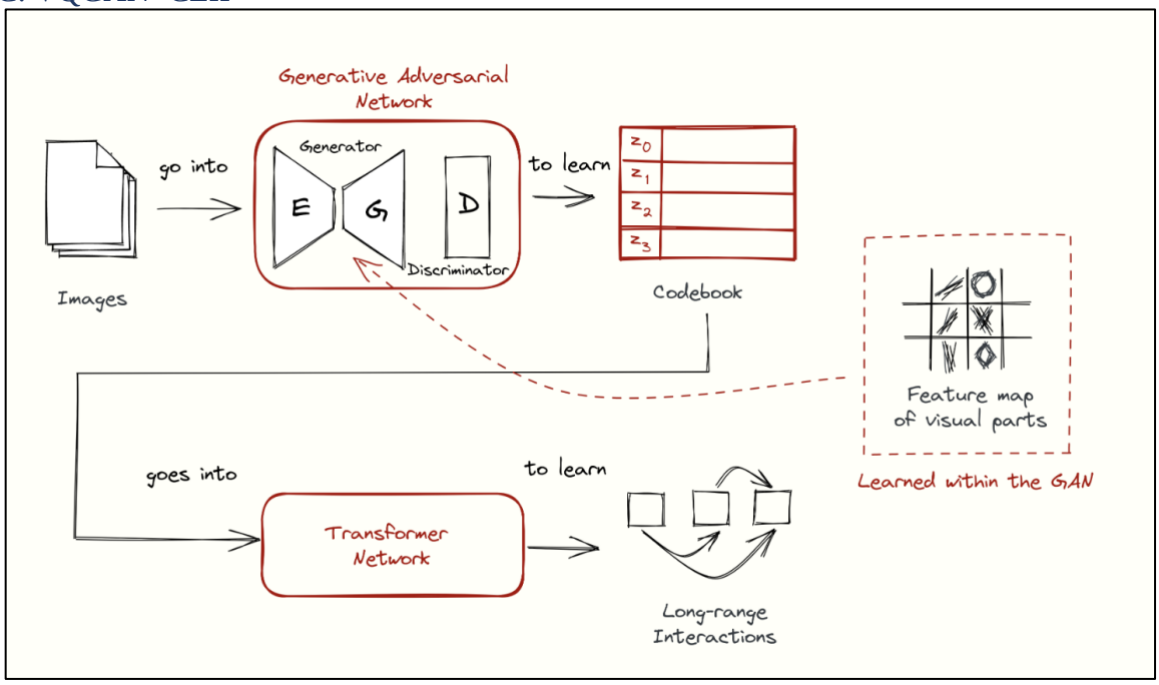

Figure 9: An overview of the VQGAN architecture (Miranda).

As I researched how people created their own text-to-image programs, I stumbled upon VQGAN+CLIP, which was made by Katherine Crowson (Crowson, Biderman and Kornis). VQGAN stands for Vector Quantized Generative Adversarial Networks (Miranda). This technique used VQGAN to generate the image, and CLIP to judge how well the image matched the text prompt, which guides the generator to produce more accurate images.

VQGAN is able to both learn the visual parts of an image and also the relationships between these parts. Typically, a convolutional neural network (CNN) can be used to learn the visual parts, and a transformer model can be used to learn the long-range dependencies. Since we want VQGAN to do both, logically we would think to combine these approaches. One way is to flatten the learned visual parts of the image into a sequence and feed it into a transformer network. However, this approach does not work as it encounters a limitation in the transformer network: its computation time squares with the length of the input sequence because of the transformer attention mechanism (Miranda).

The attention mechanism allows the model to focus on distinct parts of the input sequence, depending on their relevance to the task at hand, as the mechanism processes it. At each layer of the transformer model, the input sequence is broken down into a series of queries, keys, and values. The queries are used to compute a query score for each key, which decides how relevant that key is to the query. Then the query scores are used to compute a weighted sum of the values, where the weights are given by the softmax of the query scores. This weighted sum tells the model how much attention should be placed on each part of the input sequence.

Attention became an issue because for a 224×224px image the input would have length $224^2 \times 3$, which is way above the capacity for a GPU. A quick fix would be to reduce the context by downsampling the image to 32, 48, and 64px (Miranda). However rather than restricting the image size, VQGAN uses a codebook to represent the visual parts and employs a two-stage structure by learning an intermediate representation before feeding it into a transformer. This codebook is created by performing vector quantization (VQ), hence the VQ part of VQGAN.

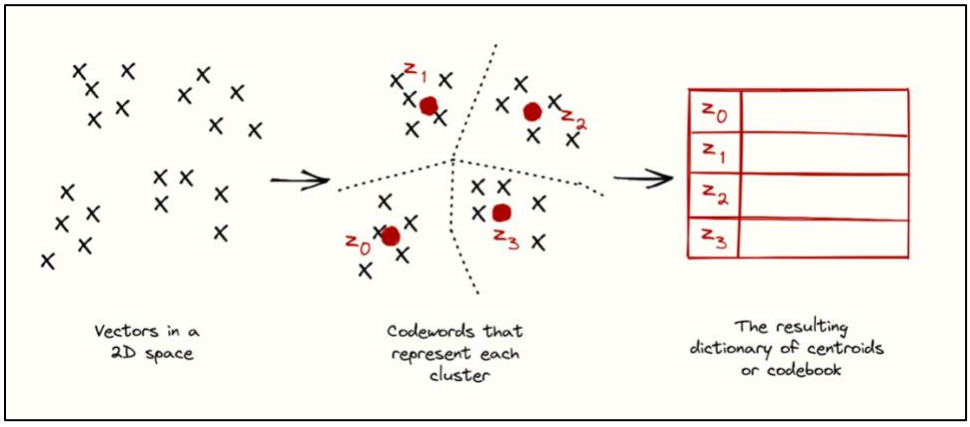

Figure 10: An illustration of vector quantization (Miranda).

Although we have discussed the use of CNNs for learning the visual parts, in VQGAN we replace the CNN with a GAN model. During the GAN training, the GAN learns to

reconstruct the input images from a low-dimensional codebook that remains fixed in size throughout this process and is learned simultaneously. The architecture of the generator in the VQGAN model follows an encoder-decoder architecture (Miranda). The theory is that if we have perfect reconstruction, then this means the encoder has found a suitable representation of the data.

In the very first round of training, the codebook is initialized with random values. At this point we can say that the codebook is likely to be a poor representation of the image, as it has not had sufficient time to learn the most representative codewords. As training progresses, the codebook refines its codewords to better represent the visual parts of the images. It also remains fixed in size, meaning the codewords themselves are updated and refined. VQGAN by itself is used to generate high-quality images based on other images (similar to NST), but when we add CLIP, it can be used to generate high-quality images based on text prompts (the architecture of VQGAN can be seen in figure 8).

The Generator: Encoder-Decoder

An image is first input into the encoder, where it is encoded into latent code (a low dimensional vector that represents the image in compressed form). This latent code is then quantized, meaning the latent code is mapped to the closest codeword (Miranda). The codeword contains information about features present in the image. The codeword is then passed to the decoder. This process can be thought of in terms of recipes and cookbooks; The codewords are recipes and the codebook is a cookbook. The decoder then accesses the codebook, using the codeword as an index, and then reconstructs the image based on the codeword.

The Discriminator

Next, the original image from the dataset and the reconstructed image are passed to the discriminator. The discriminator then calculates the adversarial loss which is in the GAN training.

Text-to-Image

In order to give VQGAN the ability to generate images based on text input, we need to combine it with a transformer model. This can be any kind of transformer model, but we will discuss the combination of VQGAN and CLIP. VQGAN stays responsible for all of the tasks it handled before, but now CLIP is introduced as the text encoder and the generated image encoder. CLIP will compare how well the generated image (from the decoder) matches the text input, using cosine similarity. This comparison is used to guide the generator to produce an image that matches the input text. In summary: CLIP will encode the text input into a fixed-length vector. The vector is then quantized using the vector quantized variational autoencoder (VQ-VAE), which works similarly to how the generator did in VQGAN. Next, the decoder part of the VQGAN generator takes the codeword and uses it to generate (or "reconstruct") an image. CLIP now takes over for the second part. The generated image is encoded by CLIP into another vector, and the similarity between the text vector and image vector is computed using cosine similarity. The model is trained to minimize the difference between the computed similarity and a target similarity score, using backpropagation and gradient descent.

My Proposal

My proposal is not an in-depth idea by any means, nor do I have any of the details of how this would work out. The idea is more on an abstract level as I do not have the experience and expertise in Python required to be able to code this proposal. During my research, I came upon a model called ArtEmis which will be discussed later in Chapter 5, section 1. My proposal is to

combine ArtEmis with VQGAN + CLIP. We know AI models can generate high quality images, but as we can see from figure IDK THE NUM, they do not always exceed our expectations when it comes to emotions. By introducing ArtEmis, which produces a vector of emotion probabilities, these text-to-image models can further improve their emotional understanding. One possible approach to combining VQGAN + CLIP with ArtEmis could involve the following steps:

- Use ArtEmis to analyze the emotional content of a given input text and generate an emotional vector which represents the desired emotional content for the image generated.
- (These next 2 steps are basically the normal function of VQGAN + CLIP). Use CLIP to encode the input text and then use the decoder from VQGAN to generate the image based on the quantized text encoding.
- Use CLIP to encode the generated image into an image encoding and compare it with the text encoding to ensure the generated image is similar to the input text.
- Use ArtEmis to analyze the emotional content of the generated image and compare it with the desired emotional vector from the first step (from the input text).
    - If the generated image is not emotionally similar to the desired emotional vector, then go through the process once again until a satisfactory image is generated.

This proposal or approach would require the modification and/or development of new models, training data and computational resources. It would also require careful integration and training of the two models to ensure they work together seamlessly. It is hard to say whether the proposed model would be better than other competing models without conducting experiments and comparing results. However, combining different models and techniques can often lead to

improvements in performance. So, the proposed model could potentially lead to a better model for generating images with emotional content.

The process of learning and understanding the technologies and programming languages necessary for developing an art generative AI based on text prompts was a challenging but rewarding journey. There were several obstacles, such as my lack of experience in Python and unfamiliarity with IDEs and libraries, but through research and help from my mentor I was able to overcome them. I was able to learn the importance of neural networks and natural language processing in these programs and gained a much deeper understanding of the CLIP model, which is crucial to most text-to-image programs, such as DALL-E and StableDiffusion. My experience showed me how much time and effort goes into developing AI systems and highlighted the importance of collaboration and teamwork in the field. Despite not being an expert in Python or AI development, my research provided me with enough understanding to be able to propose a program that would use pre-existing models to improve the emotional capabilities of art generative AI.

Chapter 5: Instability in the Community: is AI art really art?

5.1: The Debate: Is AI art "art?"

The integration of AI into art, has sparked many debates regarding the legitimacy of AI "art". Some say it is art, like a gallery in California that held an exhibition called "Artificial Imagination," while others argue that this new art form lacks originality and creativity and is nothing more than a form of plagiarism (Metz). There have been many specific events that have fueled this fire such as AI wining an art prize, or AI art selling for about half a million dollars (Quackenbush). The debates around AI-generated art touch upon topics such as emotional depth and expression, authenticity and value of art, and creativity and authorship.

5.1A: Emotional Depth and Expression

Figure 11: Achiloptas, Guibas and team's algorithm in action. Seen above are the algorithm's written justifications for its emotional categorization of the art works above the captions (Myers).

One of the debates surrounding the integration of AI into the art field, centers on the question of whether AI-generated art works can convey emotional depth and expression, given the inability of AI to experience emotions the same way humans do. On the other hand, some argue that although AI cannot experience emotions, it can be programmed to produce works that elicit emotional responses in humans. At Stanford University, a doctoral candidate Pano Achlioptas in collaboration with partners in France and Saudi Arabia, and a Stanford Engineering Professor Leonidas Guibas have worked together to create a program, ArtEmis, that captures the emotions of artists' works (Myers). The algorithm categorizes the artist's works into one of eight emotional categories, including but not limited to, awe, amusement, fear and sadness, and then justifies the emotional read (Figure 11). Although this program does not create

art that is eliciting an emotional response, it can be used in combination with other programs to generate art that will. There already exist programs like DALL-E, StableDiffusion, or Midjourney that generate images based off text prompts, these images that can elicit emotions, if the text prompt is specified (Figures 12 and 13).

Ujué Agudo, Miren Arrese, Karlos G. Liberal, and Helena Matute, conducted a study to investigate how people perceive the abilities of human artist and artificial intelligence in creating artwork. The audience was randomly put into 2 groups, AI artist or human artist. All groups watched the same video in which an AI improvised a piano melody while painting on a canvas following the rhythm of the music. After, they told AI artist group that the artist was an AI, and they told human artist group that the artist was human. Then they asked 2 questions: 1) "Now that you have seen the video of this Artificial Intelligence/ of this artist/ of these artists, to what degree would you say that it arose your emotion?" 2)"And how would you rate the artist's sensitivity?" (Agudo, Arrese and Liberal). The answers were provided using a scale system from 0-10. They correctly predicted that "people would attribute AI a poorer ability than human artist to perform with sensitivity a piece of artwork and a weaker ability to evoke emotions in the audience." (Agudo, Arrese and Liberal). Regardless of whether or not AI has emotions or can elicit emotions what really matters is if people will ever treat it like normal art.

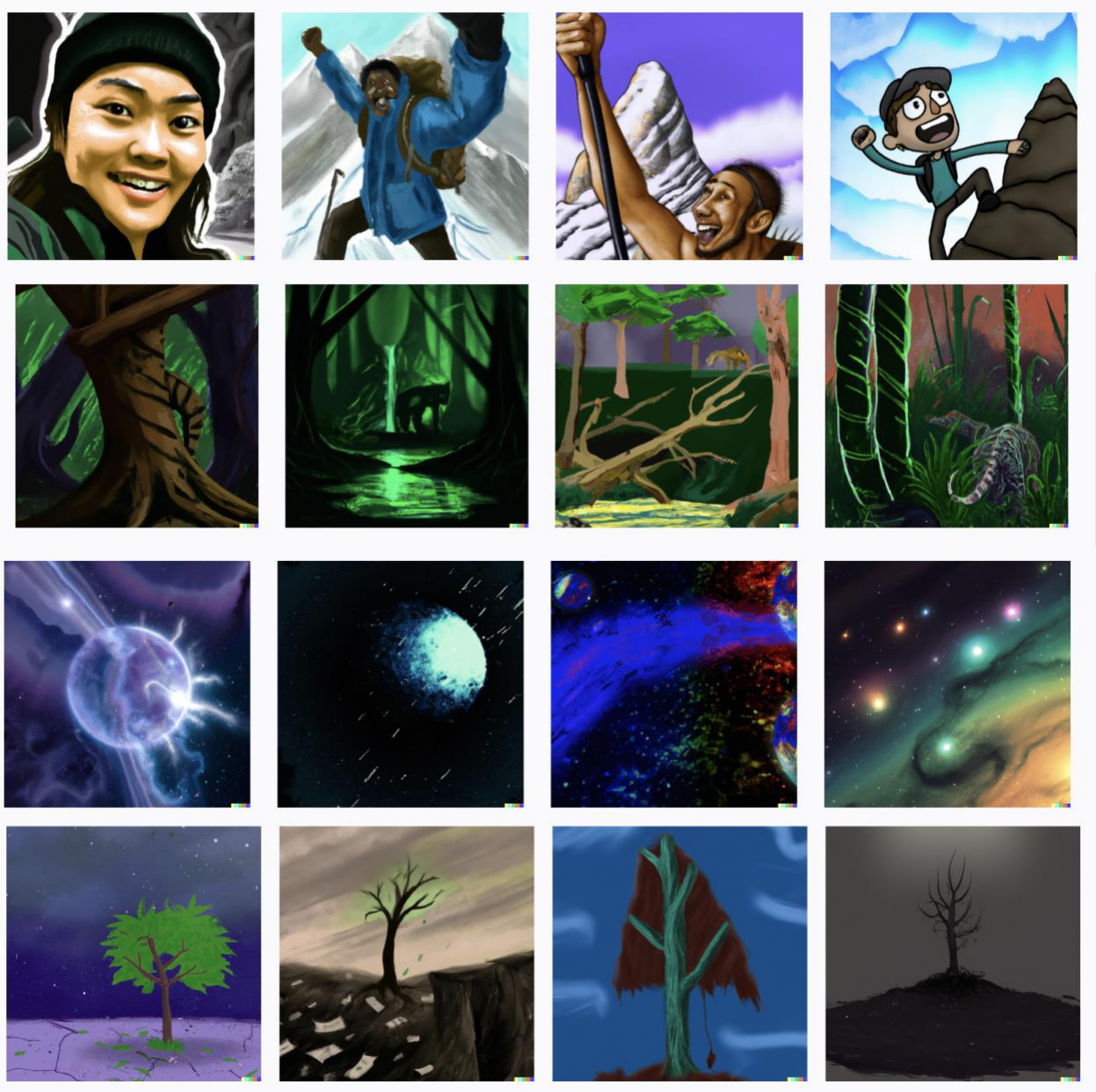

Figure 12: Images generated using DALL-E. The prompts are as follows (top to bottom): A happy digital art piece of a mountain climber A digital art piece of the jungle that elicits fear. A digital art piece of space that elicits awe. A sad digital art piece of a poverty-stricken tree.

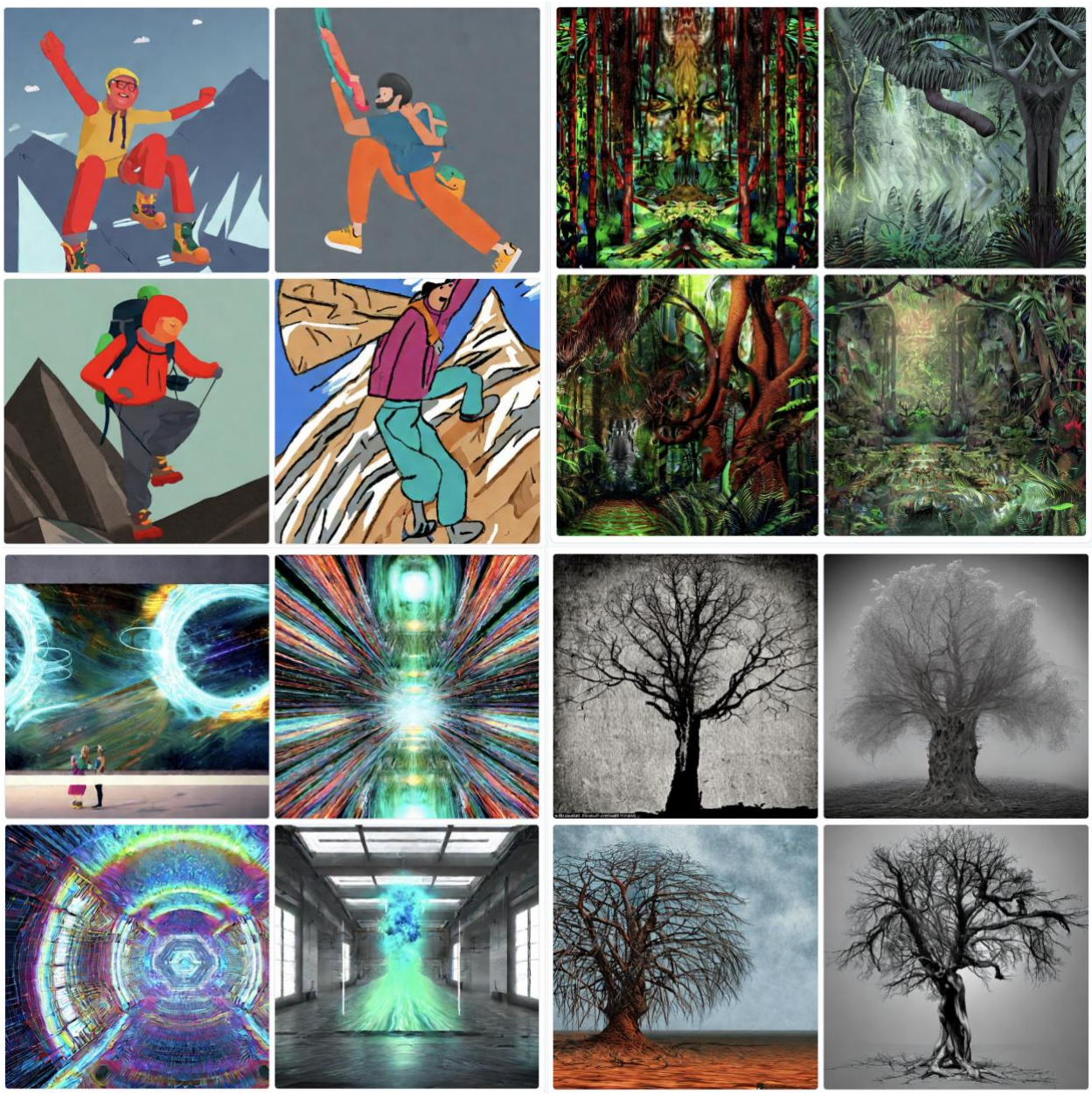

Figure 13: Images generated using StableDiffusion. The propmts are as follows (left to right then top to bottom): A happy digital art piece of a mountain climber A digital art piece of the jungle that elicits fear. A digital art piece of space that elicits awe. A sad digital art piece of a poverty-stricken tree.

5.1B: Authenticity and Value of Art

Some argue that AI-generated art lacks authenticity and value because it is created by machines rather than humans. Here, art is seen as a product of human creativity and emotions, which machine are not capable of replicating. They also argue that AI-generated art is often created for commercial use, meaning it lacks the authenticity and individuality that comes from

the personal expression of human artists. In the article “Is AI Changing the Way We Think About Authenticity?” by Carl Blaine Horton Jr., the author explores the impact of AI on the concept of authenticity in art and what it means for a work of art to be authentic. Blaine Horton Jr. uses the term “’motivational authenticity’” or “’artistic authenticity’” to describe a type of authenticity used to evaluate art. This term refers to when a work of art was or was not produced for some external reason (Blaine Horton Jr.). This concept is then further broken down into two categories, extrinsic motivation, which refers to being motivated by external factors such as money, fame or peer pressure, and intrinsic motivation, which refers to the desire to self-express. It becomes difficult to place AI programs in this new space. On one hand, one can argue that AI lacks the emotional drive to have intrinsic motivation, but one could also argue that art produced by AI is never produced by extrinsic motivation-fame, money, or respect. While the debate around the authenticity and value of AI-generated art continues, it is clear that the impact of AI on the art world is significant and important to discuss.

5.1C: Creativity and Authorship

Figure 14: Barret's statement on Twitter regarding the use of his program in the production of the piece titled "Edmond de Belamy, from La Famille de Belamy" which sold for $432,500 (Cohn).

Likely the most important and popular subject of debate, is the legitimacy or authorship of AI-generated art. In 2018 a portrait piece, titled "*Edmond de Belamy, from La Famille de Belamy*" generated by an AI program sold for $432,500 at Christie's-over 40 times the original estimate of $7,000-$10,000 (Cohn). A few days after the announcement of the auction, an artist, 19-year-old Mr. Barret, said the code he wrote and shared online was used in the production of "*Edmond de Belamy, from La Famille de Belamy*" (Cohn) (See figure 14 for Barret's statement on Twitter). The print was created by Obvious, a trio of 25-year-old French students, who did not publicize the fact that they substantially borrowed Barret's code (Vincent). On one hand the AI portrait for sale at Christie's is seen as a milestone. On the other hand, the fact that the trio borrowed Barret's code to create the portrait, and did not properly credit him, raised questions about authorship and ownership in AI-generated art, and has marred the milestone for some to an extent.

This controversy over authorship and ownership of AI-generated art has led to legal actions, such as the class-action lawsuit filed by collector and artist, Kelly McKernan. McKernan noticed their name was frequently being used in A.I.-driven image generation. Upon further investigation, on the Discord chat that runs the AI generator called Midjourney, McKernan discovered that users had used their name more than twelve thousand times in public prompts, to the point that the work produced seemed "'a little infringe-y'" (Chayka). In January 2023, McKernan joined two other artists, Sarah Andersen and Karla Ortiz, in filing a class-action lawsuit against Midjourney, StableDiffusion and DreamUp.

These kinds of cases or occurrences raise the question of what creativity means in the context of AI-generated art and whether machines can be considered creative. The concept of creativity is difficult to explain and often terms such as "inspiration" or "intuition" are used when trying to explain how one has creative ideas. Ramón López de Mántaras, in his article "Artificial Intelligence and the Arts: Towards Computational Creativity", discusses the notion that creative works and ideas are not created from nothingness but rather are influenced by historical and cultural backgrounds as well as personal experiences, "it is a fruit of the cultural inheritance and the lived experiences." (Mántaras). Mántaras proposes the idea that a creative idea is a novel and valuable combination of known ideas. In this sense, computers can be creative, as they have been given knowledge which they have acquired cumulatively. AARON is a robotic system that can pick up a paintbrush with its robotic arm and paint on a canvas on its own. It generates unique drawings of people in a botanical garden, rather than making copies of existing work. AARON may have never experienced the knowledge it has but once it understands the knowledge it has it can make use of it whenever it needs it. On the other hand, AARON can never break the "rules" it learned from the knowledge it accumulated. For example,

AARON knows a human has 2 arms and 2 legs, it understands gravity, center of balance, and occlusions, meaning it understands that if a human body is partially occluded then it might have only one arm or one leg, but when not occluded AARON always draw 4 limbs. This means AARON will never "imagine" the possibility of drawing other forms of abstraction (Mántaras). AI programs may have creativity in Mántaras form of creativity, but many other argue that for this very reason AI programs are plagiarizing other artists.

If creativity is an accumulation or combination of known knowledge, and AI is using this knowledge to generate art, shouldn't the creators of this knowledge and art be acknowledged and credited in some form? In order for an AI program to generate art it must be trained on a dataset of images, art, and/or matching text captions. Companies that gather this data are not crediting or paying the creators of the data, leading to many arguments about copyright issues. Kaloyan Chernev, founder of DDG, says that the dataset comprises "'largely public domain images sourced from the internet'", but many artists and illustrators say that these databases often include copyrighted images (Shaffi). One side argues that human art is shaped by memories, history, and cultural influences, and is enriched by personal touches. However, the opposing side argues that AI can only replicate, meaning it lacks the ability to add originality to its creations. As stated by Shaffi, in a publication featured in the Guardian, there is a perspective that AI generators operate similarly to humans in terms of being influenced by others' work. However, in the same publication, Biddulph counters this argument by highlighting that human artists infuse emotions, nuances, and incorrect memory into their creations. Biddulph explains in the following thorough example,

> "'If I'm making a painting and decide it should be Hockney-esque, I'm not going to trawl the internet for millions of Hockney-esque images, work out exactly what traits makes

> these images Hockney-esque, then apply them to my picture, systematically with forensic accuracy. I'm going to think 'I like the way Hockney juxtaposed blocks of purple, green and ochre in that painting of a field I saw at the National Gallery.' And then I'll attempt to add that to my picture. Inevitably, I'll misremember it, and will probably end up creating something that bears a faint resemblance to something Hockey once painted, but in my own style'" (Shaffi).

The question of crediting and acknowledging the creators of data and art used in AI-generated creations is a continuing debate. Many argue that if creativity is an accumulation of knowledge, then those who contributed to the dataset used for training should be compensated. Others, such as Kaloyan Chernev, founder of DDG, claim the dataset used in AI generators mostly comprise of public domain images from the internet. Furthermore, while some argue that AIs' dataset are equivalent to the human accumulation of knowledge, others contend that human artists add unique emotions, nuances, and faulty memory into their creations, setting them apart from AI-generated stuff. All things considered, the ethical and legal implications of creativity and authorship in the context of AI-generated art remains a complex and evolving discussion.

Safety

This section will be much shorter and will not be discussed in detail.

Another topic of discussion in AI-generation, is safety. Although companies like OpenAI, have put in place content policies, there are still plenty of other AI art generative programs that may not have these rules (Open AI). Kaloyan Chernev admitted that during the initial launch of Text 2 Dream, people tried to "'generate images of nude children, despite the fact that no such images were present in the training dataset'" (Shaffi). Most recently in January of 2023, Twitch streamer QTCinderella discovered her face had been deepfaked onto a porn performer's body (Farokhmanesh). As reported by Farokhmanesh, the images first caught attention when viewers

of Brandon “Atrioc” Ewing’s stream spotted a website on his screen that contained nonconsensual deepfake pornography he’d bough of popular streamers, like QTCinderella, Pokimane, and Maya Higa. These images may not be real, but to those unaware of the situation, to their knowledge they are real. Victims are harassed with explicit videos and images made in to look like them. Families and friends, of these victims may not have the online knowledge to understand that the media has been falsified. Deepfakes can be used as a weapon and are used predominantly against women. In 2018, Rana Ayyub, a journalist based in Mumbai, was the victim of a deepfake attack after writing a critical article about India’s ruling party, BJP (Jaiman). Her face was superimposed on a porn video, she was then doxed, and the video was distributed on social media. The harassment and humiliation sent Ayyub to the hospital with heart palpitations and led her to withdraw her online presence (Jaiman). Stories like Ayyub’s and QTCinderella’s are not uncommon, and with AI advancing rapidly, ensuring safety should be a great concern.

There are several debates surrounding AI-generated art, including its ability to convey emotional depth and expression, its authenticity and value, and its legitimacy or authorship. Some argue that because AI is a machine it cannot express emotions, but others argue it can be programmed to do so. The question of whether AI-generated art is “art”, is also a topic of debate, given AI’s inability to experience life the way humans do. Authorship is the greatest and blurriest topic of debate, as it becomes increasingly to navigate AI-generated art rights in the legal system. As AI continues to work its way into more creative fields, these topics will continue to be debated and new issues will arise.

Works Cited

A.I For Anyone. *node (computer science)*. n.d. March 2023. <https://www.aiforanyone.org/glossary/node#:~:text=A%20node%20is%20a%20point%20in%20a%20network%20where%20data,represent%20relationships%20between%20the%20data.>.

Abadi, Martín, et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." 2015. Online Tutorial.

Agudo, Ujué, et al. *Assessing Emotion and Sensitivity of AI Artwork*. 5 April 2022. March 2023. <https://www.frontiersin.org/articles/10.3389/fpsyg.2022.879088/full>.

AICAN. *Link Between Heaven and Earth*. AICAN, Miami.

Alammar, Jay. "The Illustrated Transformer." 27 June 2018. *github.* Blog. November 2022. <https://jalammar.github.io/illustrated-transformer/>.

Allen, Greg. "CDAO-references." April 2020. *Chief Digital and Artificial Intelligence Office.* 12 January 2023. <https://apps.dtic.mil/sti/pdfs/AD1099286.pdf>.

Blaine Horton Jr., Carl. *Could AI Change Way We Think about Authenticity?* July 23 2017. March 2023. <https://act-lab.gsb.columbia.edu/research-projects/could-ai-change-way-we-think-about-authenticity>.

Carew, Joseph M. *In-depth guide to machine learning in the enterprise*. February 2023. March 2023. <https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning#:~:text=Reinforcement%20learning%20is%20a%20machine,learn%20through%20trial%20and%20error.>.

Cetinic, Eva and James She. "Understanding and Creating Art with AI: Review and Outlook." 18 February 2021. *arXivLabs.* 03 October 2022.

Chayka, Kyle. *Is A.I ART STEALING FROM ARTISTS?* 10 February 2023. March 2023. <https://www.newyorker.com/culture/infinite-scroll/is-ai-art-stealing-from-artists>.

Cohn, Gabe. *AI Art at Christie's Sells for $432,500*. 25 October 2018. March 2023. <https://www.nytimes.com/2018/10/25/arts/design/ai-art-sold-christies.html>.

Copeland, B.J. *Artifial intelligence*. 11 November 2022. 12 January 2023. <https://www.britannica.com/technology/artificial-intelligence>.

Coursera. *Deep Learing vs. Machine Learning: Beginner's Guide*. 22 March 2023. March 2023. <https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide>.

Crowson, Katherine, et al. "Vqgan-clip: Open domain image generation and editing with natural language guidance." *17th European Conference on Computer Vision (ECCV)*. Tel Aviv, Israel: Springer Nature Switzerland, 2022.

Education, IBM Cloud. *What is Artificial Intelligence (AI)?* 03 June 2020. 26 September 2022. <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>.

Elgammal, Ahmed, et al. "CAN: Creative Adversarial Networks, Generating “Art” by Learning About Styles and Deviating from Style Norms." 21 June 2017. *arXivLabs.* 23 September 2022.

Farokhmanesh, Megan. *The Debate on Deepfake Porn Misses the Point*. 1 March 2023. April 2023. <https://www.wired.com/story/deepfakes-twitch-streamers-qtcinderella-atrioc-pokimane/>.

Gandharv, Kumar. *Top 5 applications of Convolutional Neural Network*. 29 June 2022. March 2023. <https://indiaai.gov.in/article/top-5-applications-of-convolution-neural-network>.

Gatys, Leon A., Alexander S. Ecker and Matthias Bethge. "Image Style Transfer Using Convolutional Neural Networks." *@016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. 2016. 2414-2423.

George, Benjamin St. and Alexander S. Gillis. *Tech Accelerator: A guide to artifical intelligence in the enterprise*. January 2023. 12 January 2023. <https://www.techtarget.com/searchenterpriseai/definition/Turing-test#:~:text=The%20Turing%20Test%20is%20a,cryptanalyst%2C%20mathematician%20and%20theoretical%20biologist.>.

Goodfellow, Ian J., et al. "Generative adversarial networks." *Communications of the ACM* 63.11 (2020): 139-144.

Google. *Colaboratory*. n.d. March 2023. <https://research.google.com/colaboratory/faq.html>.

Google Developers. "Machine Learning." *Overview of GAN Structure*. 18 July 2022. Online Course.

History of Data Science. *Ian Goodfellow: MAchine Learning Wunderkind*. 03 June 2021. March 2023. <https://www.historyofdatascience.com/ian-goodfellow-machine-learning-wunderkind/>.

Hutto, C.J and E.E Gilbert. "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text." *Eight International Conference on Weblogs and Social Media (ICWSM-14)*. Ann Arbor, MI, 2014. Github Repository.

IBM. *A Computer Called Watson*. n.d. March 2023. <https://www.ibm.com/ibm/history/ibm100/us/en/icons/watson/>.

—. *What is deep learning?* n.d. March 2023. <https://www.ibm.com/topics/deep-learning>.

Jaiman, Ashish. "Deepfakes Harms & Threat Modeling." *Medium* 19 August 2020. Blog.

Mántaras, Ramón López de. "Artificial Intelligence and the Arts: Towards Computational Creativity." n.d. *OpenMind BBVA.* March 2023. <https://www.bbvaopenmind.com/en/articles/artificial-intelligence-and-the-arts-toward-computational-creativity/>.

Metz, Rachel. *Is AI art really art? This California gallery says yes*. 20 November 2022. March 2023. <https://www.cnn.com/2022/11/20/tech/ai-art-exhibit-ctpg/index.html>.

Miranda, LJ. *The Illustrated VQGAN*. 8 August 2021. March 2023. <https://ljvmiranda921.github.io/notebook/2021/08/08/clip-vqgan/>.

Mordvintsev, Alexander, Christopher Olah and Mike Tyka. *Inceptionsim: Going Deeper into Neural Networks*. 18 June 2015. March 2023. <https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>.

Myers, Andrew. *Artist's Intent: AI Recognizes Emotions in Visual Art*. 22 March 2021. March 2023. <https://hai.stanford.edu/news/artists-intent-ai-recognizes-emotions-visual-art>.

Open AI. *Content policy*. 19 September 2022. April 2023. <https://labs.openai.com/policies/content-policy>.

Quackenbush, Casey. *A Painting Made by Artificial Intelligence Has Been Sold at Auction for $432,500*. 26 October 2018. March 2023. <https://time.com/5435683/artificial-intelligence-painting-christies/>.

Radford, Alec, et al. "Learning Transferable Visual Models From Natural Language Supervision." 26 February 2021. *Cornell University.* <https://arxiv.org/abs/2103.00020>.

Ramesh, Aditya, et al. "Hierarchical Text-Conditional Image Generation with CLIP Latents." 13 April 2022. *arXivLabs.* 15 09 2022.

Roose, Kevin. *An A.I.-Generated Picture Won an Art Prize. Artists Aren't Happy.* 2 September 2022. March 2023. <https://www.nytimes.com/2022/09/02/technology/ai-artificial-intelligence-artists.html>.

Shaffi, Sarah. *'It's the opposite of art':why illustrators are furious about AI*. 23 Jan 2023. March 2023. <https://www.theguardian.com/artanddesign/2023/jan/23/its-the-opposite-of-art-why-illustrators-are-furious-about-ai>.

Singh, Aditya. "How does DALL-E 2 Work?" Augmented Startups, 27 April 2022. Blog.

Tao, Feng. "A New Harmonisation of Art and Technology: Philosophic Interpretations of Artificial Intelligence Art." *Critical Arts* 36.1-2 (2022): 110-125.

Vincent, James. *How three French students used borrowed code to put the first AI portrait in Christie's*. 2018 October 2018. March 2023. <https://www.theverge.com/2018/10/23/18013190/ai-art-portrait-auction-christies-belamy-obvious-robbie-barrat-gans>.

Wang, Justin Jay. *OpenAI*. 5 Jan 2021. March 2023.