A Proposed Approach for AI Doodle Generation with a Hybrid Intelligent Agent

Prof. Dyaneshwar Bavakar[1], Prof. Ramesh Shahabade[2], Vishal V. Shinde[3], Bhargav V. Modak[4], Manas N. Telavane[5], Shantanu Parameswaran[6]

[1,2,3,4,5,6]Computer Engineering Department, Terna Engineering College, Mumbai, India

Abstract

Doodles are a form of human expression and communication that capture the essence of concepts and emotions in a simple and intuitive way. However, generating doodles from natural language descriptions is a challenging task that requires both understanding the meaning and context of the input, and producing the appropriate strokes and shapes of the output. In this paper, we present our approach to tackle this problem. Our proposed system leverages the vector-based nature of doodles and the semantic relationships of words. We propose use of a dataset of arrays of coordinates as drawn by humans (stored as vectors) to train our agent. There are several mechanisms that enable the agent to handle different aspects of the task, such as input word filtering, semantic mapping and keyword weighing, finding out closest words to input in the agent's dictionary using vector embeddings, running one or multiple deep learning models in parallel depending on the number of objects, using the semantic mappings to train the assembler to linearly translate, transform or scale objects as per their semantic positional relationship, and using the turtle to draw the objects on a canvas. We identify and outline possible benchmarks, tests to evaluate the performance and quality of the agent using various metrics and user feedback. We also discuss the potential applications and implications of such an agent, such as art and design, education, and therapy.

Impact Statement: The field of text-to-doodle generation is currently under-explored, with limited research and no end-user product. Existing datasets are specific and lack the flexibility to expand based on user feedback. Our research aims to bridge this gap by proposing a system that not only generates doodles from text but also mimics human-like stroke patterns. We are innovating ways to decompose complex shapes into simpler ones and vice versa, and to generate doodles even in the absence of exact matches in the database. Our approach also includes semantic mapping to maintain proper positional relationships when placing multiple doodles on a canvas. The successful implementation of our proposed architecture will revolutionize the way we interact with digital art platforms, providing a more intuitive and engaging user experience. It holds potential for wide-ranging applications in education, entertainment, and beyond, offering a novel way of visual communication and expression.

Index Terms: Artificial intelligence, Doodle Generation, Hybrid intelligent systems, Intelligent systems, Multi-agent systems, Natural language processing, Neural networks, Semantic search, Vector images.

I. INTRODUCTION

FREE-HAND Doodles are a unique form of human expression, capturing the essence of thoughts, ideas, and emotions in a simple yet profound manner. Unlike meticulously crafted drawings or paintings, doodles are spontaneous and free-flowing, often created subconsciously while the mind is otherwise engaged. They are the rawest form of human creativity, unbound by the constraints of precision or realism.

Doodles are distinct from other forms of hand-drawn drawings in several ways. Firstly, they are typically created without a preconceived plan or design [1], making each doodle a unique and unrepeatable manifestation of the moment. Secondly, doodles are often abstract and symbolic, rather than literal representations of the physical world. This gives them a certain fluidity and flexibility, allowing for a wide range of interpretations [2].

Doodles are not only a form of artistic expression, but also a powerful tool for learning and communication. Research has shown that doodling can enhance memory, attention, and comprehension [3], as well as stimulate creative thinking and problem-solving [4]. Doodling can also serve as a visual cue, a graphical representation of a concept or a message that can facilitate understanding and recall. For example, doodling can help students remember key points of a lecture, or help speakers illustrate their ideas to an audience.

Interestingly, doodling is also closely related to language, the most fundamental mode of human communication. Many language scripts, such as Chinese, Arabic, and Devanagari, have evolved from pictographic or ideographic symbols that resemble doodles [5]. Doodles share a common trait with ancient writing systems like Egyptian hieroglyphs [6] and Chinese characters : they both represent ideas or concepts through visual symbols.

Moreover, doodling can be seen as a form of visual language, a system of signs and symbols that can convey meaning and information [7]. For instance, doodling can be used to create logos, icons, diagrams, and charts that can communicate complex or abstract ideas in a simple and intuitive way. This symbolic representation, free from the constraints of any specific language or writing system, embodies abstract thought in a spontaneous, subconscious form.

However, doodling is not a universal language. Just as writing systems evolved differently across the globe, so too did the way people doodle or draw everyday objects. Different cultures may have different styles, preferences, and associations when it comes to doodling. The way a person in one geographical region draws a circle or a chair can be distinctly different from how someone in another region does [8]. Some cultures may favor geometric shapes, while others may prefer organic forms. Some cultures may use doodles to express emotions, while others may use them to convey facts. Some cultures may have specific meanings or connotations for certain doodles, while others may interpret them differently [9]. Therefore, understanding the cultural context and diversity of doodling is essential for creating and appreciating doodles. These geographical differences in doodling styles offer an intriguing perspective on human cognition, communication, and creativity.

In recent years, advances in artificial intelligence (AI) have enabled the generation of realistic and diverse visual content from natural language descriptions. These AI systems, such as GPT-4 [10] and Gemini [11], use large-scale neural networks to learn from multi-modal data, such as text, images, videos, and audio, and produce various types of media output. These AI-generated visual cues can have various applications, such as advertising, entertainment, education, and art.

However, most of these AI systems are not specialized or optimized for generating doodles. They tend to produce rasterized images that are pixel-based and high-resolution, rather than stroke-based and low-resolution. They also tend to lack the human-like creativity and spontaneity that characterize doodles. Moreover, they may not account for the cultural differences and nuances that influence doodling. Therefore, there is a need for a novel AI system that can generate doodles from text in a human-centric and culturally-aware manner.

A. Motivation

The main motivation for this research is to propose an AI system that can generate doodles from text, following the strokes and patterns of human doodlers. Such a system would have several benefits and applications, such as:

- It would provide a fun and engaging way for users to interact with AI and explore their creativity. Users could input any word or phrase and see how the AI would doodle it, or they could collaborate with the AI to create co-doodles. Users could also share their doodles with others and compare the different styles and interpretations of the AI and human doodlers.
- It would enable the collection and analysis of large-scale data on how different cultures associate certain phrases with certain doodles. By allowing users from different regions and backgrounds to use the system, we could gather valuable insights into the cultural diversity and similarity of doodling. This could help us understand the underlying cognitive and social processes that shape doodling, as well as improve the cross-cultural communication and appreciation of doodles.
- It would advance the state-of-the-art of AI in generating visual cues from natural language. By designing and implementing a system that can generate doodles from text, we will face and overcome several technical challenges, such as modeling the stroke dynamics and structure of doodles, incorporating the cultural context and preferences of users, and evaluating the quality and diversity of the generated doodles.

To the best of our knowledge, this is the first research project that aims to propose a system that can generate and handle multi-object doodles from text. Previous attempts at demonstrating the possibility of text-to-doodle generation [12] have been explained in more detail later. We believe that this proposal has the potential to make a significant contribution to the fields of AI, doodling, and visual communication. In the following sections, we will describe the related work, the proposed agent architecture, possible methods to test and benchmark the agent, and the ethical implications of this project.

B. How the rest of the paper is organized

The rest of the paper is presented as follows. In Section II, we will provide a comprehensive background and introduce key terms relevant to the field of doodling and AI. Section III will review the related work on generating visual cues from natural language, and highlight the gaps and limitations of the existing approaches. Section IV will describe the proposed Doodle Drawing Agent agent, a possible AI system that could generate doodles from text, following the strokes and patterns of human doodlers. Section V will present the experiments one can use to evaluate the performance and quality of the Doodle Drawing Agent agent, using various metrics and user feedback. Section VI will conclude the paper and summarize the main contributions and outcomes of this research, and suggest some possible future work and applications of the Doodle Drawing Agent agent, and explore the ethical and social implications of this research.

II. Background and Key Terms

This section provides an overview of key terms and concepts related to text-to-sketch generation, laying the groundwork for a detailed exploration of the literature and cutting-edge models. We will begin by discussing doodles and how they differ from technical drawings or engineering sketches. Then we will delve into other aspects, including raster images, vector graphics, text-to-image generation, text-to-doodle generation, and AutoGPT. Table I presents the different acronyms and abbreviations used in this text.

A. Doodles

A doodle is a type of drawing created spontaneously, without the use of tools or references. Unlike technical drawings, illustrations, fine art, professional sketches, forensic sketches, cartoons, or oil paintings, which prioritize accuracy, visual appeal, or specific artistic styles, free-hand sketches prioritize the freedom of expression and the exploration of ideas. These sketches capture the essence of a subject rather than focusing on precise proportions, colors, textures, or perspectives.

TABLE I ACRONYMS AND ABBREVIATIONS

Acronym	Definition
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
CGAN	Conditional Generative Adversarial Network
GPT	Generative Pre-trained Transformer
LSTM	Long short-term memory
Seq2Seq VAE	Sequence to Sequence Variational Autoencoder
MMD	maximum mean discrepancy

Doodles require no formal training or specialized equipment. Anyone can create them, regardless of their artistic skills. This accessibility and simplicity make doodles a unique and inclusive form of artistic expression, allowing individuals to convey creativity and ideas without limitations.

In this paper, the term “sketch” specifically refers to free-hand drawings or doodles, emphasizing their spontaneous and expressive nature. By understanding the distinct characteristics and accessibility of free-hand sketches, we can explore their significance and potential applications in various fields.

B. Raster Images

An image, in the context of this paper, refers to a pixel-based raster representation that encompasses various types of visual content. These include:

- Photographs: Captured by cameras to faithfully represent scenes. These may be film cameras or digital cameras. In the case of the former, the images may be scanned to obtain a digital representation of the photograph.
- Computer-generated images: Created by computer algorithms or artificial intelligence.

Raster images are constructed using tiny rectangular pixels (picture elements) arranged in a grid formation. Each pixel contains information about color, intensity, and other visual attributes. Raster editing tools like GIMP or Adobe Photoshop allow users to modify and manipulate these images. Common editing features include cropping, resizing, retouching, color correction, and visual effects.

The term “image” broadly covers both photographs and computer-generated visuals. Regardless of their origin or modification, pixel-based raster representations play a crucial role in various fields. Understanding the challenges and techniques related to working with pixel-based visual content is essential for advancing research in this area.

C. Vector Graphics

Vector graphics, unlike raster images, are not constructed from a grid of pixels. Instead, they are composed of paths, which are defined by a start and end point, along with other points, curves, and angles along the way. A path can be a line, a square, a triangle, or a curvy shape. These paths can be used to create simple drawings or complex diagrams. Paths are also used to define the characters of specific typefaces.

While raster images are resolution-dependent – meaning they cannot scale up to an arbitrary resolution without loss of apparent quality – vector images are defined in terms of points on a Cartesian plane and they are resolution-independent. This means they can be scaled to any size and printed at any resolution without losing detail or clarity. As a result, vector graphics are more versatile for certain types of tasks than raster graphics.

In the context of doodles, vector graphics offer a significant advantage. The stroke-based nature of doodles aligns well with the path-based nature of vector graphics. Each stroke in a doodle can be represented as a path in a vector graphic, capturing the sequence and direction of the drawing process. This allows for a more authentic replication of the human drawing process, as opposed to the static representation offered by raster graphics.

Moreover, vector graphics allow for easy manipulation of individual elements of the doodle. Simple shapes can be combined to create complex figures, or complex shapes can be broken down into simpler components. This aligns with the spontaneous and exploratory nature of doodling, where the artist iteratively adds, removes, or modifies elements of the drawing.

D. Text to Image Generation

Text-to-image generation, a burgeoning domain within the realm of artificial intelligence, leverages advanced models to transform textual descriptions into visual content. This technology has witnessed remarkable progress, with Conditional Generative Adversarial Networks (CGANs) at the forefront. CGANs, which utilize recurrent neural networks (RNNs) or transformer models to encode text descriptions, guide the image generation process, significantly enhancing the quality and diversity of generated images.

Innovative models such as Control GPT [20] have emerged to augment the controllability of text-to-image generation models. This advancement allows for the creation of photo-realistic images that adhere to specific textual instructions, thereby addressing the limitations of previous models. DALL-E 3 and Midjourney are notable examples of models that excel in generating high-quality images from textual prompts, with DALL-E3's capability to produce diverse images from a single prompt showcasing its versatility.

Imagen by Google Brain [21] and Stable Diffusion models employ diffusion processes to generate images. These models initiate with a random image and iteratively refine it to match the given text prompt, demonstrating their proficiency in generating high-resolution images.

Despite these advancements, challenges persist. The "uncanny valley" effect, where generated images are almost but not quite realistic, and the difficulty in generating images of hands, which are complex and highly variable structures, remain significant hurdles. Additionally, generating text within images is a complex task due to the intricacies of fonts, styles, and backgrounds.

The field of text-to-image generation continues to evolve, with researchers exploring new methods and models to overcome these challenges and further enhance the quality and diversity of generated images.

E. Text to Doodle Generation

Text to doodle generation represents a fascinating intersection of artificial intelligence and creative expression, leveraging textual prompts to generate unique, hand-drawn illustrations. The input to text-to-doodle generation models typically consists of textual descriptions, which serve as the creative seed for the doodle. The output format is a visual representation, often in the form of a digital image or a vector graphic, capturing the essence of the textual input in a hand-drawn style. This process involves converting textual prompts into a sequence of strokes, which are then rendered into a visual form.

The conversion of raw data into a format suitable for training models like SketchRNN involves the use of .ndjson and .npz files. The .ndjson files contain newline-delimited JSON objects, which represent the strokes of a doodle in a sequence. Table II specifies the format of each ndjson object. These files are then converted into .npz files, which store the stroke data in a compressed format optimized for machine learning models. This conversion process is crucial for preparing the data for training, ensuring that the model can learn from the stroke sequences to generate doodles.

AARON [13] is a notable model in the field of text-to-doodle generation. It utilizes a sophisticated algorithm to interpret textual prompts and generate corresponding doodles. AARON's approach to doodle generation is groundbreaking, as it combines the power of deep learning with the nuanced understanding of human creativity to produce unique and artistic doodles.

SketchBird [14], while innovative in its use of AI to generate doodles, does not strictly adhere to the traditional definition of doodles. SketchBird focuses on creating digital illustrations based on textual prompts, which, while visually similar to doodles, differ in their digital nature and the absence of the hand-drawn quality that defines doodles. This distinction is important as it highlights the evolving nature of doodle generation, where the line between traditional and digital art is blurring.

CLIPDraw [15], another model in the text-to-doodle generation space, does not follow a temporal sequence in generating doodles. Unlike models that generate doodles by sequentially adding strokes based on the textual input, CLIPDraw generates doodles in a non-linear fashion. This approach allows for a more abstract interpretation of the textual prompts, leading to a wide range of creative outputs. However, this method may result in less coherent or less predictable doodles compared to models that adhere to a temporal sequence. CLIPDraw is capable of achieving success with ambiguous input in multiple ways and reliably producing drawings that look like modern art. This indicates that while the technique might not be ideal for emulating childlike doodles due to its tendency towards complexity, it might be quite adept at interpreting and creating abstract art similar to Picasso's works.

SketchRNN [16] is a pivotal model in the development of text-to-doodle generation, known for its ability to generate doodles based on a sequence of strokes. This model has been instrumental in advancing the field, demonstrating the potential of AI to mimic the creative process of human doodling. In the next main section, we will explore SketchRNN in more detail, along with variations and evolutions like Pixelor [17].

III. Related Work

This section will delve into the existing body of work related to generating doodles from text, focusing on datasets, models, and their applications. The aim is to provide a comprehensive overview of the current state of research in this domain, highlighting the datasets available for training and evaluation, as well as the models developed to tackle the challenge of generating doodles. We will look at available datasets which contain stroke data in both .ndjson or .npz format, then look at SketchRNN, the seminal work in text-to-doodle generation. We will also look at Pixelor, an offshoot based on SketchRNN that is optimized for Pictionary, and look at some possible models which can be used for automated evaluation of generated doodles.

A. Available Datasets

1. **QuickDraw Dataset:** The QuickDraw Dataset is a vast collection of 50 million drawings across 345 categories, contributed by players of the game "Quick, Draw!". These drawings are captured as timestamped vectors, enriched with metadata such as the drawing prompt and the country of origin. This dataset is particularly valuable for training models to recognize and generate doodles, offering a rich resource for researchers and developers in the field of machine learning and computer vision.
2. **TUBerlin:** The TUBerlin Dataset [22] has a specific focus on the analysis of non-expert portrayals of everyday objects such as an apple and a fan. The primary objectives include assessing human recognition accuracy, comparing it with computational methods, and developing a computational recognition model. The dataset crafted by the authors comprises 20,000 sketches, meticulously categorized into 250 distinct object classes. The chosen categories are exhaustive, recognizable based on shape alone, and specific enough to enable accurate recognition.
3. **UJI Pen characters:** Another resource for researchers and practitioners in computer science, the dataset comprises handwritten character samples collected from 60 writers across two different sites and phases. The character set includes letters (both uppercase and lowercase), digits, and punctuation marks. Each sample is represented as a sequence of (x, y) coordinates, capturing the pen's trajectory during writing. The dataset facilitates research in handwriting recognition, pattern recognition, and machine learning algorithms. Researchers can explore variations in writing styles, inter-writer differences, and the impact of different writing tools.

B. SketchRNN

SketchRNN [16] stands as a pivotal milestone in the intersection of neural networks and artistic expression. This model introduces stroke-based drawings of common objects and was developed as part of the Magenta project, an initiative exploring the role of machine learning in creating art and music. Magenta, initiated by researchers and engineers from the Google Brain team, aimed to develop new deep learning and reinforcement learning algorithms for generating songs, images, drawings, and other creative materials.

Magenta encompasses both Python-based libraries and TensorFlow.js for browser-based applications. It provides pretrained models, documentation, and instructions on training custom models with user datasets. SketchRNN, as a creation of Google Brain, is included in the Magenta repositories, with its pretrained models and comprehensive documentation. SketchRNN employs a sequence-to-sequence Variational Autoencoder (Seq2Seq VAE), with a bidirectional Long Short-Term Memory(LSTM) serving as the encoder. This model undergoes training to reconstruct the original stroke sequences while simultaneously ensuring a normal distribution across the elements in the latent space.

Both the encoding and the decoder's sampling mechanism operate stochastically, resulting in unique and varied sketches. The model's ability to generate both conditional and unconditional sketches allows for a wide range of outcomes, from simple doodles to more complex drawings.

The Seq2Seq VAE, consists of two main parts: an encoder and a decoder. The encoder takes a sketch (represented as a sequence of pen strokes) as input and compresses it into a fixed-size vector in the latent space, also known as the 'z'-vector. The decoder, on the other hand, takes this 'z'-vector and reconstructs the original sketch. The model is trained to minimize the difference between the original sketch and the reconstructed sketch, which is measured by a loss function. The bidirectional LSTM in the encoder allows the model to capture dependencies in both directions of the sequence, thereby better understanding the structure and style of the sketch.

However, SketchRNN has certain limitations. For example, it struggles with handling long sequences and fails to encode necessary contextual information.

While SketchRNN remains an integral part of Magenta, it's essential to note that as of April 2023, the Magenta project, along with other Google Brain initiatives, was suspended following the merger of Google Brain into Google DeepMind.

C. Pixelor

Pixelor [17], like SketchRNN, is also a sequence-to-sequence model. However, it uses a Transformer as the encoder, which alleviates the long-term dependency problem of RNNs, thereby capturing better contextual information. Pixelor also employs different metrics such as Wasserstein Distance and Maximum Mean Discrepancy (MMD) instead of Log-likelihood and KL divergence to match the latent distribution to a prior.

The Transformer encoder in Pixelor allows the model to handle longer sequences than SketchRNN, thanks to its self-attention mechanism that allows each input in the sequence to interact with all other inputs, irrespective of their distance. This helps the model to capture long-term dependencies in the data, which are often crucial for understanding complex sketches. The use of Wasserstein Distance and MMD offers a more principled way to compare distributions, which helps in improving the quality of the generated sketches.

Pixelor is used for the task of playing against human players in pictionary, hence is optimized for drawing in such a way that optimizes the recognition time of a particular doodle, rather than draw it in any natural way that a human would draw. It provides insights into how human culture and temporal sequence in drawing may affect their capacity to win at pictionary.

D. SketchR2CNN

SketchR2CNN [18] presents a significant advancement in the domain of text-to-doodle generation by capitalizing on the temporal ordering and grouping information inherent in freehand sketches. Unlike conventional methods that rasterize sketches into binary images for classification, SketchR2CNN adopts a novel single-branch attentive network architecture. This architecture, named RNN-Rasterization-CNN, or Sketch-R2CNN for brevity, seamlessly integrates recurrent neural networks (RNNs) and convolutional neural networks (CNNs) to fully exploit the dynamic nature of sketches for recognition.

SketchR2CNN operates by taking a vector sketch with grouped sequences of points as input. It employs an RNN for stroke attention estimation in the vector space and a CNN for 2D feature extraction in the pixel space. To bridge the gap between these two spaces within the neural network framework, SketchR2CNN introduces a neural line rasterization module. This module converts the vector sketch, along with the attention estimated by the RNN, into a bitmap image, which is subsequently processed by the CNN. Crucially, the neural line rasterization module is designed in a differentiable manner, enabling seamless end-to-end learning.

The key innovation of SketchR2CNN lies in its ability to leverage the dynamics of sketches through an attention mechanism, leading to more robust recognition performance compared to state-of-the-art methods. By preserving the temporal ordering and grouping information of sketches, SketchR2CNN achieves superior accuracy in recognizing sketched objects.

IV. The Doodle Drawing Agent

A. Overview

The proposed drawing agent embodies several distinct features essential to its functionality:

- it leverages a vast repository of predefined words and strokes sourced from diverse datasets, including the widely used Quickdraw dataset. These resources serve as foundational elements for generating doodles corresponding to user input;
- it ensures the integrity and appropriateness of generated content, our agent incorporates the Obscenity npm package. This package employs pattern matching and transformers to identify and filter out obscene words and phrases from user input, thereby upholding standards of decency and suitability;
- it exhibits adaptability in its operational modes, dynamically adjusting its behavior based on the availability of pre-trained models for specified prompts. This feature enhances the agent's versatility and responsiveness to varying input conditions, optimizing its performance across different contexts;
- it includes provisions for user feedback mechanisms, facilitating continuous refinement and expansion of its underlying database. By incorporating user input, the Doodle Drawing Agent iteratively enhances its dataset, thereby improving the quality and diversity of generated doodles over time.

B. Auto-GPT and Hybrid AI

To propose a Doodle Drawing Agent, we draw inspiration from groundbreaking methodologies such as AutoGPT [19], which epitomizes the concept of Hybrid AI—where disparate AI components seamlessly collaborate to achieve emergent intelligence. AutoGPT represents a paradigm shift in AI research, demonstrating how a confluence of independent systems, orchestrated by a central thread, can manifest behavior that transcends the capabilities of individual components.

AutoGPT operates by harnessing the capabilities of multiple AI models, such as GPT-4, GPT-3.5, or GPT-3, and orchestrating their interactions to tackle diverse tasks. At its core, AutoGPT employs a hierarchical approach, where a main thread, typically running on powerful hardware, delegates specific sub-tasks to smaller instances of GPT models deployed on different hardware. Each sub-task is tailored to the strengths of the respective model, leveraging their unique capabilities to accomplish a particular aspect of the overall task.

The concept of AutoGPT can be elucidated through the Chinese Room analogy, wherein a person inside a room with no understanding of Chinese can produce coherent responses to Chinese input by following predefined rules. Similarly, each AI model within AutoGPT operates as a self-contained entity, processing input and generating output based on learned patterns and algorithms. While individual models lack understanding or consciousness, their collective interactions give rise to emergent behaviors that mimic intelligent responses.

Drawing parallels with the human brain and emergent behavior observed in nature, such as the coordinated movements of ants in colonies, an agent could decomposes the drawing task into granular sub-tasks, each assigned to a specialized agent.

What could such tasks be? One might think of the following sub-tasks involved in creating an intelligent system for text to doodle generation:

1. **Input Handling and Semantic Mapping:**

- The agent receives a textual prompt, which could describe a scenery or scenario involving multiple objects.
- Semantic mapping may involve analyzing the prompt to identify key objects, attributes, and relationships, potentially aiding in subsequent tasks.

2. **Searching for Closest Words in Database of Vectors or Strokes:**

- For each object identified in the prompt, the agent searches a database of vectors or strokes to find the closest representations.

- Utilizing clues from semantic analysis, the agent refines its search to identify relevant visual elements corresponding to each object.

3. **Parallel Sketching DL Models:**

- The agent invokes separate sketching deep learning models for each identified object, enabling parallel processing and specialization.
- Each model is tasked with generating strokes or sketches specific to its assigned object, leveraging its unique capabilities and training data.

4. **Assembling and Transforming Individual Drawings:**

- The agent assembles individual strokes or sketches outputted by the DL models into coherent drawings.
- Understanding the spatial relationships between objects, the agent applies linear transformations to adjust position, size, orientation, and other attributes as necessary.

5. **Using a Canvas to Draw the Shapes in the Pipeline:**

- The agent utilizes a canvas or drawing environment to render the assembled drawings.
- Each drawing is placed on the canvas according to its spatial attributes, ensuring proper positioning and alignment within the scene.

In envisioning an ideal agent capable of performing these tasks, we recognize the complexity and interdependence of each component in the text-to-doodle generation process. By systematically decomposing the task into granular sub-tasks and allocating specialized agents to handle each aspect, the agent exhibits emergent behavior reminiscent of coordinated actions observed in natural systems. Through seamless collaboration and orchestration, such an agent could potentially emulate human-like creativity and ingenuity in transforming textual prompts into expressive doodles.

This paper proposes a posssible model and architecture actualize this ideal doodle drawing agent. Like AutoGPT, the Doodle Drawing Agent embodies a decentralized architecture where multiple agents collaborate to achieve a common goal. Each agent corresponds to a specific task within the drawing process, ranging from input handling to stroke generation and canvas rendering. Despite their autonomy, these agents synchronize their actions under the guidance of a central orchestrator, akin to the conductor of an orchestra. With the Doodle Drawing Agent, we strive to propose a system that leverages emergent behavior to generate doodles from text inputs with creativity and ingenuity.

A. Preprocessing

In creating such an agent, the different mini-agents and tasks necessitate distinct datasets and preprocessing methodologies tailored to the specific requirements of each mini-agent:

1. Input text and Semantics

For this module, the input format comprises a string, encompassing various forms such as sentences, paragraphs, single words, or even empty inputs. Although compatible with

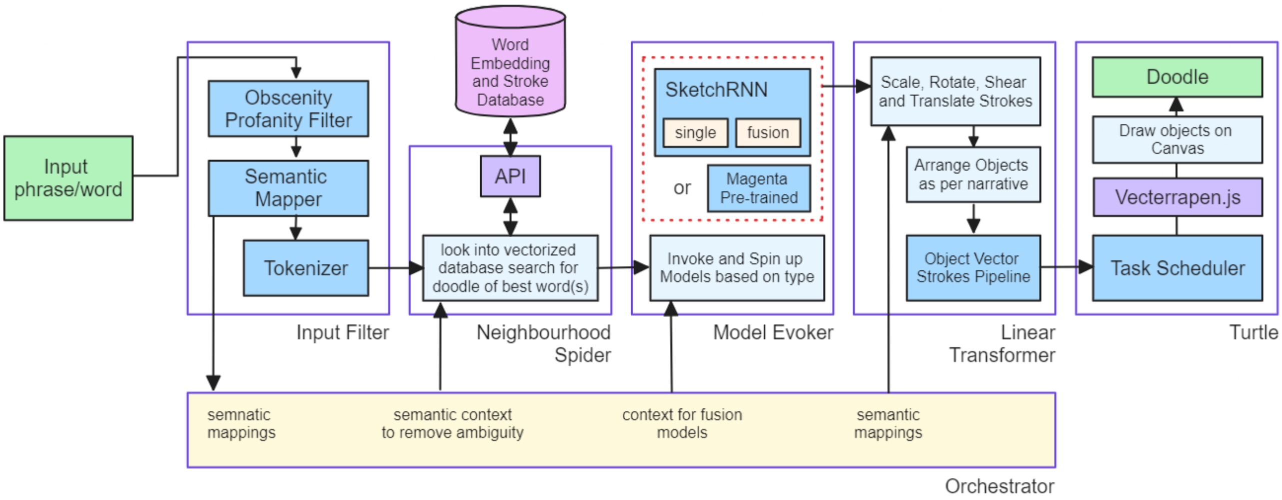
Input phrase/word
Obscenity Profanity Filter
Semantic Mapper
Tokenizer
Input Filter
Word Embedding and Stroke Database
API
look into vectorized database search for doodle of best word(s)
Neighbourhood Spider
SketchRNN
single
fusion
or
Magenta Pre-trained
Invoke and Spin up Models based on type
Model Evoker
Scale, Rotate, Shear and Translate Strokes
Arrange Objects as per narrative
Object Vector Strokes Pipeline
Linear Transformer
Doodle
Draw objects on Canvas
Vecterrapen.js
Task Scheduler
Turtle
semnatic mappings
semantic context to remove ambiguity
context for fusion models
semantic mappings
Orchestrator

Fig. 1. Architecture of the Doodle Drawing Agent

multiple languages, the scope of this module and paper is delimited to English. The output of this task entails semantic mapping or a collection of words representing drawable objects within the prompt. To facilitate semantic mapping, we curated a dataset associating phrases containing objects with corresponding semantic mappings. For instance, the phrase 'cat in a tree' is mapped to { minor: 'cat', rel: 'inside', major: 'tree' }.

2. Neighborhood Search

In this module, the input format comprises individual objects represented by singular words, occasionally accompanied by modifiers to disambiguate homonyms. Leveraging the QuickDraw dataset augmented with the Magenta-specific dataset, we execute a meticulous search operation. The Magenta dataset, utilized by SketchRNN, is a modified iteration of the QuickDraw dataset. Due to hardware constraints, we extract the top 5 strokes for each object from the QuickDraw dataset. The transformed dataset is then uploaded to Weaviate, a vectorized database service offering vector embeddings and facilitating neighbor word searches based on vector similarity. This setup enables an API to retrieve an object along with its top 5 vector strokes, yielding a comprehensive dataset for subsequent processing.

3. Deep Learning Models

In this module, the input format comprises the object along with its closest words and their corresponding top 5 strokes. Minimal preprocessing is necessitated, with specific consideration given to scenarios involving a singular closest word versus multiple closest words. The output entails a singular vector stroke array for each object, facilitating seamless integration into the subsequent stages of the drawing process.

B. Architecture of the Agent

The Doodle Drawing Agent uses 6 modues, or mini-agents, for performing the text-to-doodle generation task. A visual representation of this architecture can be seen in Figure 1.

1. Semantic Mapper

The Semantic Mapper module encompasses input handling, filtering, and semantic mapping functionalities. Upon receiving a sentence from the user, it employs logic-based null input filtering and language filtering, restricting operations to English language inputs. Profanity filtering is executed using the Obscenity npm library, which employs NLP pattern matching and transformers to flag obscene words. The module outputs a list of words with positional semantic relationships and stores semantic mappings in a global state for subsequent utilization by the Orchestrator. Additionally, individual objects are passed to instances of the Neighbourhood Spider module for further processing.

2. Orchestrator

The Orchestrator module serves as a central entity, overseeing the operation and coordination of all independent agents within the Doodle Drawing Agent agent. It stores the semantic context and facilitates its provision to any module upon request, ensuring seamless interaction and collaboration among disparate components.

3. Neighborhood Spider

The Neighborhood Spider module interfaces with the Weaviate vectorized database to retrieve information about drawable objects obtained from the Semantic Mapper module. It traverses the database to obtain direct matches or closest matches based on vector similarity. In cases where a pretrained Magenta model exists, the module returns the word with an empty strokes array. Otherwise, it provides the word (or group of words) along with their top 5 vector strokes in an array format. The Neighborhood Spider is exposed through an API on Weaviate.

4. Model Evoker

The Model Evoker module orchestrates the scheduling and invocation of deep learning models, such as Sketch RNN, or pretrained models from Magenta, in parallel. It coalesces multiple strokes into a single stroke or aggregates strokes based on input specifications. The module dynamically selects and invokes different models depending on whether the input corresponds to an exact match or an aggregate match.

5. Linear Transformer

The Linear Transformer module interfaces with the Orchestrator to obtain semantic mappings, which are utilized to manipulate and transform drawings. It facilitates operations such as scaling up, scaling down, orientation adjustment, rotation, shear, and arrangement of different objects within the drawing. Scheduled drawings are subsequently queued for processing within the pipeline.

6. Turtle

The Turtle module renders drawings onto a browser interface. Due to the absence of a native implementation of the turtle library, Vecterrapen.js, an SVG graphics library on npm, is used for drawing operations.

5. EXPERIMENTS

In this section, we outline potential benchmarks and experiments designed to evaluate the performance of each component of the proposed Doodle Drawing Agent. These experiments serve as a guideline for future researchers or developers who implement the model.

a. Semantic Mapper

1. Input:

- Did the Semantic Mapper handle empty inputs?
- How did it handle inputs containing only numeric values?
- Was it able to process special characters appropriately?
- How did it handle foreign language inputs?
- Did the Semantic Mapper effectively filter out profanities?

2. Semantic:

- Did it generate accurate mappings for basic shapes like triangles, squares, etc?
- How well did it handle manually selected shapes from the dataset?

3. Word Neighborhood Spider

- Did the Word Neighborhood Spider retrieve the most suitable word representations for different inputs?
- How did varying parameters, such as the limit on words, affect its performance?
- Was the module able to understand abstract concepts?
- How did it perform with very similar, somewhat similar, and completely different words?

4. Models

- Were exact word matches accurately drawn by the Models module?
- How well did fusion models perform in generating drawings?
- What was the recognizability of the drawings evaluated through human assessment?
- How satisfied were users during beta testing with the drawings generated?
- Were the outputs recognizable by another AI, such as Sketch R2CNN?
- What improvements were observed when using an adversarial network to enhance stroke aggregation?
- At what point did a singular model break under stress testing with a high number of strokes?
- What was the threshold at which the entire deployment of models failed under stress testing?

5. Linear Transformer

- Did rotation, translation, shearing, and scaling operations function as intended?
- How well did the Linear Transformer handle operations such as unions, intersections, and mutual exclusivity?
- How accurately did it position drawings on the canvas?

6. Turtle

- Were all drawings rendered correctly on the screen by the Turtle module?
- What was the reliability of the rendering process across multiple trials?
- How did the time required for rendering more complex drawings vary?

These benchmark questions serve to validate and refine each component in the Doodle Drawing Agent agent, ensuring its effectiveness and robustness in transforming textual prompts into expressive doodles.

CONCLUSION

In conclusion, our paper proposes a possible architecture for a sophisticated Doodle Drawing Agent capable of transforming textual prompts into expressive doodles. Through the meticulous design and integration of various components such as the Semantic Mapper, Word Neighborhood Spider, Models, Linear Transformer, and Turtle, such an agent can demonstrate promising capabilities in generating visually coherent representations from textual input.

Moving forward, several avenues for future research and development present themselves. One promising direction involves exploring boolean operations with vector representations to enable the creation of more complex drawings. Additionally, the prospect of generating a new dataset by crowdsourcing user-drawn doodles holds potential for enriching the diversity and expressiveness of the agent's output.

Furthermore, a reinforcement learning-based approach offers an intriguing avenue for refinement, where user ratings of final results could inform the weighting of database entries and strokes, thereby enhancing the agent's performance over time. Such iterative improvements could significantly enhance user satisfaction and the overall usability of the Doodle Drawing Agent.

Beyond its immediate applications, the agent holds promise for various educational and practical applications, including the generation of graphs, flowcharts, and other instructional materials. By leveraging the power of AI to translate text into visual representations, the agent stands to revolutionize the way individuals interact with and conceptualize information.

In essence, our work represents a significant step towards bridging the gap between text and visual representation, opening up exciting possibilities in a different direction than existing established raster image generation models.

References

1. Xu, P, Hospedales, TM, Yin, Q, Song, Y-Z, Xiang, T & Wang, L 2023, "Deep Learning for Free-Hand Sketch: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 285-312.
2. Qutub, Afnan. (2012). Communicating Symbolically: The Significance of Doodling between Symbolic Interaction and Psychoanalytical Perspectives. 8. 72-85.
3. Sundararaman, Deekshita. “Doodle Away: Exploring the Effects of Doodling on Recall Ability of High School Students.” International Journal of Psychological Studies (2020).
4. Khalid, M., Saad, S., Abdul Hamid, S. R., Ridhuan Abdullah, M. ., Ibrahim, H., & Shahrill, M. (2020). Enhancing creativity and problem solving skills through creative problem solving in teaching mathematics. Creativity Studies, 13(2), 270-291.
5. Rita Borgo, Johannes Kehrer, David H S Chung, Min Chen, "Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications", May 2013.
6. Karthik Variath Divakaran, Shiyas Ahamed S, Tushar Renji Thoonkuzhy, Yadul Manoj, Mr. Ajith S, "A Sustainable Decipher of Egyptian Hieroglyphs", vol.10, no: 5, pp. 827-834, May 2023.
7. Watson, Benjamin. (May 2008). Oodles of doodles? “Doodling behaviour and its implications for understanding palaeoarts.” Rock Art Research. 25. 35-60.
8. Restoy, S., Martinet, L., Sueur, C., & Pelé, M. (2022). Draw yourself: How culture influences drawings by children between the ages of two and fifteen. Frontiers in Psychology, 13. Retrieved from https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2022.940617. DOI: 10.3389/fpsyg.2022.940617. ISSN: 1664-1078.
9. Wang, Y., & Cui, X. (2015). A Study on Cultural Connotation of Animal Words in English and Chinese. In Proceedings of the 2015 International Conference on Education, Management and Computing Technology (pp. 57-60). Atlantis Press. DOI: 10.2991/icemct-15.2015.14
10. OpenAI et al. (2024). GPT-4 Technical Report. arXiv:2303.08774 [cs.CL].
11. Gemini Team et al. (2024). Gemini: A Family of Highly Capable Multimodal Models. arXiv:2312.11805 [cs.CL].
12. Cheng, Mingyong. 2022. "The Creativity of Artificial Intelligence in Art" Proceedings 81, no. 1: 110.
13. Cohen, P 2016, "Harold Cohen and AARON", AI Magazine, vol. 37, no. 4, pp. 63-66.
14. Shaozu Yuan, Aijun Dai, Zhiling Yan, Zehua Guo, Ruixue Liu, Meng Chen; Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, 2021, pp. 2443-2452.
15. Frans, K., Soros, L. B., & Witkowski, O. (2021). CLIPDraw: Exploring Text-to-Drawing Synthesis through Language-Image Encoders. arXiv:2106.14843 [cs.CV].
16. Ha, David R and Douglas Eck. “A Neural Representation of Sketch Drawings.” ArXiv abs/1704.03477 (2017).

17. Ayan Kumar Bhunia, Ayan Das, Umar Riaz Muhammad, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, Yulia Gryaditskaya and Yi-Zhe Song "Pixelor: a competitive sketching AI agent. So you think you can sketch?", ACM Transactions on Graphics, Volume 39, Issue 6, November 2020.
18. Li, Lei et al. “Sketch-R2CNN: An Attentive Network for Vector Sketch Recognition.” ArXiv abs/1811.08170 (2018).
19. Yang, H., Yue, S., & He, Y. (2023). Auto-GPT for Online Decision Making: Benchmarks and Additional Opinions. CoRR, abs/2306.02224..
20. Zhang, T., Zhang, Y., Vineet, V., Joshi, N., & Wang, X. (2023). Controllable Text-to-Image Generation with GPT-4. arXiv:2305.18583 [cs.CV].
21. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., & Norouzi, M. (2022). Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. arXiv:2205.11487 [cs.CV].
22. Eitz, M., Hays, J., & Alexa, M. (2012). How Do Humans Sketch Objects? ACM Transactions on Graphics (Proc. SIGGRAPH), 31(4), 44:1–44:10.