Storytelling through Image Narration

A Senior Honors Thesis

Submitted in Partial Fulfillment of the Requirements for

Graduation in the Honors College

By Stacey Do

Computer Science and Mathematics Major

The State University of Brockport

May 17, 2024

Thesis Director: Ning Yu, Associate Professor, Department of Computing Sciences

Abstract

The project aims to develop an interactive storytelling chatbot by integrating Google Vision API and OpenAI API, using natural language processing and deep learning algorithms. Google Vision API extracts diverse features from images, including landmarks, objects, colors, text, and logos, while OpenAI API processes this information to generate concise image descriptions. This collaboration seamlessly fuses visual understanding and language generation, creating a compelling storytelling chatbot that offers users an immersive narrative experience. The goal is to explore methods for synchronizing image analysis and natural language processing, enhancing education, entertainment, creativity, and accessibility. The project showcases the potential of cutting-edge API technologies in meaningful ways, demonstrating the continuous growth in AI advancements. The presentation will overview the process of how this has been done but also emphasize the potential of its impact and purpose to help improve education systems.

Table of Contents

Introduction.....5

Related Works.....6

a. Overview.....6
b. Image Recognition….....6
c. Natural Language Processing and Text Generation….....6
d. Combining Image Recognition and Text Generation…….....7
e. Challenges and Opportunities….....7
f. Conclusion….....8

Method….....8

a. Google Vision API…….....9
b. OpenAI API…….....11
c. Additions and Tools…….....13
d. Overview…….....14

Implementations…….....14

a. System Configuration…….....14
b. Image Input.....….....14
c. Image Processing (Google Vision).....15
d. Features Extraction (OCR)…….....15
e. StoryTelling Narration (OpenAI API)…….....15
f. Side implementations.....16

Results and Display.....17

Discussion.....19

a. Successes.....19
b. Drawbacks.....20
c. Improvements.....20
d. Future Work.....20

References.....22

Acknowledgments.....24

I. Introduction

Artificial Intelligence (AI) has impacted society over the years. Starting as an idea at the Dartmouth Conference in the 1950s, AI has evolved into a global phenomenon, growing in various aspects of daily life and not just within the STEM field. The study and use of AI have exponentially grown, leading to its implant into real-world applications such as healthcare, education, advertising, and social media platforms. These continuous advancements in AI drive technological progress and innovation.

One notable area of AI development is image recognition, which includes technologies such as optical character recognition (OCR). Image recognition involves algorithms that extract data from images, converting raw data into useful information. This capability reduces the need for manual data entry and enables AI to perform tasks traditionally done by humans. Current implementations of image recognition in everyday applications include fingerprint scanning, face detection, and voice recognition (Shidaganti et al., 2023).

Despite these advancements, a significant problem remains the integration of sophisticated AI models with image recognition to create more versatile and intelligent systems. While there are many existing models for tasks such as face detection, text scanning, and CAPTCHA solving, there is a lack of implementations combining powerful AI tools with image recognition for more unique applications.

The thesis addresses this gap by exploring the innovative use of OpenAI APIs alongside image recognition technologies. While others have developed various image recognition models, the integration of Google Vision Cloud APIs with OpenAI's advanced language models to perform tasks such as image narration has not been thoroughly explored. (Hairiri, 2023) This

approach aims to experiment with the capabilities of image recognition systems, not just purely recognizing what is in an image, but being able to provoke creative thinking from an image.

The current stage of this problem involves the widespread use of separated image recognition and language models. By combining these technologies, the research aims to demonstrate the potential for more intelligent and context-aware AI systems that can provide richer and more detailed information from visual data.

II. Related Works

a. Overview

The integration of image recognition and text generation technologies opens new possibilities for creating storytelling narratives from images. This section reviews key studies and developments in image recognition, natural language processing (NLP), and their combined use.

b. Image Recognition

Image recognition technology advanced with the development of deep learning, particularly Convolutional Neural Networks (CNNs). AlexNet, introduced by Krizhevsky, Sutskever, and Hinton (Krizhevsky, Sutskever, & Hinton, 2012), marked a breakthrough in image classification. Architectures such as VGGNet (Simonyan & Zisserman, 2014) and ResNet (He, Zhang, Ren, & Sun, 2016), further improved accuracy and capabilities. The Google Vision API uses these advancements to provide robust image analysis, including object detection, scene understanding, and optical character recognition (OCR) (Google Cloud, 2023).

c. Natural Language Processing and Text Generation

The field of NLP has also seen transformative progress with the advent of large language models like GPT-3 and GPT-4, developed by OpenAI. (Brown et al. 2020) demonstrated GPT-3's ability to generate coherent and contextually relevant text based on diverse inputs. These models have been utilized for various applications, including automated content creation, customer service, and educational tools. The OpenAI API facilitates access to these models, enabling the generation of narratives and other textual outputs from structured prompts (OpenAI, 2023).

d. Combining Image Recognition and Text Generation

Integrating image recognition with text generation to create storytelling narratives involves analyzing the image, identifying key elements, and generating a cohesive narrative. Shidaganti, Sanjana, Shubeeksh, Raman, and Thakshith (Shidaganti et al., 2023) explored the use of robotic process automation and OCR with ChatGPT to retrieve information from images, highlighting the potential for automated content creation from visual inputs. This approach underscores the feasibility of using image analysis to inform text generation.

Hariri (Hariri, 2023) provided a comprehensive exploration of ChatGPT's applications, including its limitations and future work. The study emphasized the model's potential in creative applications, such as generating narratives based on image inputs. This aligns with integrating Google Vision API and OpenAI API for storytelling.

e. Challenges and Opportunities

While combining these technologies offers innovation, there are still impediments. Ensuring image recognition accuracy is crucial, as errors can lead to

misleading information. Moreover, generating engaging and contextually appropriate stories requires sophisticated NLP models capable of understanding and creatively interpreting image content. Ethical considerations, such as the potential for bias in both image recognition and text generation, must also be addressed (Bender et al., 2021).

f. Conclusion

The reviewed literature highlights significant advancements in both image recognition and text generation technologies. Given the significant growth in OpenAI and extensive usage of ChatGPT, utilizing the company's API provides potential discoveries and usages, By integrating Google Vision API and OpenAI API, this research aims to explore new tactics for creating storytelling narratives from images and leveraging the strengths of both technologies.

III. Method

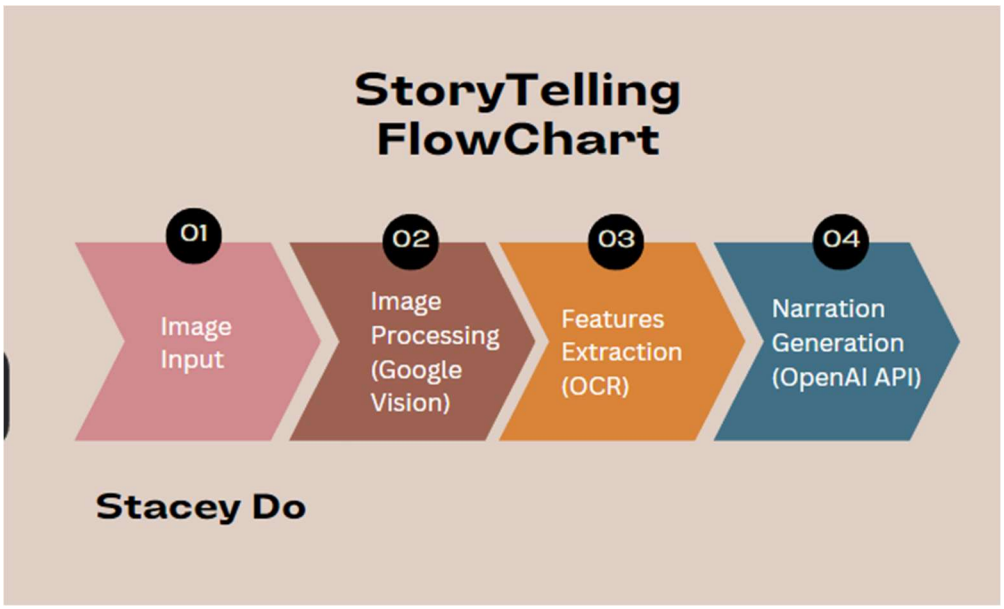

Figure 1: Flowchart of carrying out the Thesis, more in-depth process of each step in Implementations Section

Figure 1 is a diagram for executing this Thesis. The layout of Figure 1 in developing the script will begin with APIs. Based on Figure 1 above, the Google Vision APIs will first be executed to extract visual features from the image input. This is followed by the API using OCR to produce results and pass them into the OpenAI API to use its latest GPT model, GPT-4.

APIs or Application Programming Interfaces, are crucial in modern software development, enabling different systems to communicate and share data seamlessly. They retrieve information through clients like websites and allow developers to integrate functionalities from platforms such as YouTube, Google, OpenAI, Amazon, and Bing into their applications. APIs standardize requests and responses, facilitating connections to services like payment gateways and social media. This fosters innovation and efficiency, allowing developers to build sophisticated solutions that enhance user experiences. Below is a diagram of the overall structure of using APIs.

A. Google Vision API

A popular API that utilizes optical character recognition is Google Vision API. This API contains various forms of object detection ranging from location, facial, color, text, and objects in an image. This API contains multiple methods that analyze each aspect of an image. The diagram below is an example of the API being utilized to scan objects in the picture.

Figure 2: The image being used is a popular café in Paris.

```
os.environ['GOOGLE_APPLICATION_CRDENTIALS'] = "SAT.json"

device = "cuda" if torch.cuda.is_available() else "cpu"
model, transform = clip.load("ViT-B/32", device=device, jit=False)

client = vision.ImageAnnotatorClient()

FILE_NAME = 'ParisCafe.jpg'
FOLDER_PATH = r'D:\ImageRecognition\VisionAPI_Demo\Images\FamousLocations'


with io.open(os.path.join(FOLDER_PATH, FILE_NAME),'rb') as image_file:
    content = image_file.read()
Image = types.Image(content=content)

response = client.text_detection(image=Image)
response1 = client.landmark_detection(image=Image)

texts = response.text_annotations
text_read(texts)
```

Figure 3: Python script used to process photos and print out the text shown in the image.

	locale	description
0	fr	Cafe de Flore\nCAFE DE FLORE\nIGNA\nco\nStarsy
1		Cafe
2		de
3		Flore
4		CAFE
5		DE
6		FLORE
7		IGNA
8		co
9		Starsy

Figure 4: Output of Figure 2's image being processed within the code snippet in Figure 3

In the figures above, a sample image from the internet was of a famous café in France called "Café de Flores" in Figure 2. A Python script was made that processes the image using

the Google Vision API to scan text from the image in Figure 3. Figure 4 is the output where the API lists the text it can see from the image.

Google Vision also provides various other detection features associated with its API such as landmark, facial, object, color, etc. This API is an optical character recognition model for image scanning. This is one of t14he essential components in this thesis as it extracts important features from an image.

B. OpenAI API

OpenAI is a research company focused on the development of artificial intelligence. The initial idea behind the creation of this group was to research the catastrophe or potential misuse of AI which several investors pooled at least $1 billion into the company for research. OpenAI follows a non-profit structure but has hefty funding from prominent investors such as Microsoft which is the largest stakeholder for OpenAI. Popular tools such as ChatGPT, DALL-E 2, Whisper, and its API were all made by OpenAI. Many chatbots across platforms such as Lowe's, a home improvement chain, utilize OpenAI's GPT model to create virtual assistants to their advantage (Coursera, 2024).

The API possesses many different types of capabilities for users to tinker around. Most users use programming languages such as JavaScript or Python to experiment with the API. For this project, the main programming language in this thesis is Python which will be discussed in-depth later in this paper. OpenAI API can perform text and image generation, speech-to-text, and fine-tuning for pre-trained models (Nantasenamat, 2023).

```python
def generate_description():

    headers = {
      "Content-Type": "application/json",
      "Authorization": f"Bearer {api_key}"
    }
    payload = {
    "model": "gpt-4-vision-preview",
    "messages": [
        {
            "role": "user",
            "content": [
                {"type": "text", "text": f"Make a story about a father and his daughter at a park?"},
                ],
        }
    ],
    "max_tokens": 300
    }
    response = requests.post("https://api.openai.com/v1/chat/completions", headers=headers, json=payload)

    print(response.json()['choices'][0]['message']['content'])
    print("Story has been made")
    return response.json()['choices'][0]['message']['content']
```

Figure 5: A simple Python script made by me utilizing the OpenAI API, specifically the GPT model GPT-4-vision-preview

Output from GPT-4 model:

On a bright, sunny afternoon, a father and his daughter made their way to the neighborhood park. The father, a tall, sturdy man with a gentle smile, held his little girl's hand as they walked through the entrance. His daughter, a lively five-year-old with big, curious eyes, skipped alongside him with excitement.

As they approached the playground, the daughter let go of her father's hand and ran towards the swings. "Push me, Daddy!" she squealed, as she climbed onto the swing. The father chuckled and obliged, gently pushing the swing higher and higher. The daughter's laughter filled the air as she soared through the sky, feeling the wind on her face. She felt like she could touch the clouds.

After a while, they moved on to the slide. The father stood at the bottom; arms open wide, ready to catch his daughter as she slid down with a gleeful shriek. She landed safely in his arms, and he spun her around before setting her down. Her joy was infectious, and the father's heart swelled with love for his little girl.

They spent the afternoon playing on the seesaw, climbing the monkey bars, and running around the grassy field. The father watched with pride as his daughter fearlessly explored and played. He joined in on the fun, chasing her around and tickling her until she was out of breath with laughter.

Figure 5 uses the OpenAI API in Python which enables the GPT model to produce a story of a father and his daughter at a park. Based on the output, the model creates a heart-warming story and answers the prompt as I asked. This API will be the component in producing the storytelling aspect.

C. Additions and Tools

Despite the benefits of using APIs, libraries in multiple programming languages are accessible and are there when needed. For this thesis, given the first stage of this thesis was to pass in image input, there is a library in Python known as base64 which tentatively allows image URLs to be converted to an applicable format such as JPEG or PNG to then be passed along as input. Utilizing this package will allow an endless number of online images to be experimented with, allowing a great dataset and samples to choose from and make. In addition, there are other packages such as geocoder which allow coordinates to be passed to output more detailed information about a specific landmark. These packages will be further discussed later in the paper.

Python is a powerful scripting language used by developers and computer science students. The prime interest of this language is its simplicity in performing machine learning, artificial intelligence, and data science tasks. It possesses numerous APIs, data science, graphing, and machine learning packages that stand out from other programming languages because they are easy to use and accessible (Scribd, 2024). Since we are using APIs to carry out image narration, we will use Python and its packages OpenAI, pandas, geocoder, base64, and google. cloud, and Tkinter.

D. Overview

After reviewing the Google Vision and OpenAI APIs, they will be crucial in producing an adequate story-telling script. The structure of the script is given that the Google Vision API can output features of an image, it can be passed along as parameters into OpenAI's GPT-4 model to produce a story-telling initiative based on the attributes output by Google Vision API. This approach was very doable and based on its outcome, could be used in assisting education and accessibility systems.

V. Implementation

A. System Configurations

Before implementing any code into an IDE, it is important to double-check system configurations are adequate before running into any system errors. Users must have their own API key from OpenAI and Google Vision. After they obtain their API keys, they can store them as environmental variables in their system settings or can make a JSON file to set variables to them as well. In addition, users need to have at least Python 3.10 in their current configuration because past versions of Python do not support the latest Python packages for OpenAI. These are the basic essentials users need to have ready before diving into beginning the steps in Figure 1.

B. Image Input

Initially, the image was stored in a local drive and hardcoded such that the script only worked with images from the local drive. To counter the flaw of hard-coding, a function is passed where a user can grab an image URL off the internet and the function will convert the image URL to a base-64 image to be passed on to the rest of the APIs. This function allows for more versatility and allows access to unlimited images to be tested from web servers. The Python packages I used were base64 and encode.

C. Image Processing

As stated, Google Vision API contains many features it can extract from an image. To collect as many visual features as possible from an image, a function was made for each feature the API possessed. Since all the features follow the same format of submitting an image, encoding the image, and then performing the task, redundancy was prevented by simply passing the image and encoding the image as parameters instead of rewriting the same lines of code in each function. The naming conventions for these functions started with 'detect_' and then the feature it detected like text, landmark, object, etc.

D. Features Extraction (OCR)

Google Vision API is popularly known for its OCR capabilities in extracting data. For each detection, the API uses OCR algorithms to segment the image into individual parts and, use classification to find distinctive features from the part. For example, regarding text detection, Google's model captures the text regions into separate lines. Afterward, it then converts them to characters and words. By passing the image to each function, we collect substantial information from the image.

E. Narration Generation (OpenAI API)

After creating the functions in a structured manner from the Google Vision API, the OpenAI API was next implemented into the script. The API contains many different GPT models to choose from. However, the newer the GPT model, the more accurate it is in carrying out tasks compared to newer versions. Therefore, we will use the latest GPT model in their API system, GPT-4-vision-preview. Using this GPT model, we can create text prompts that pass in the outputs and visual features from the Google Vision API into the model. The model can take in multiple

text prompts at once. The first text prompt will be, “Make a story based on these image features”, and the rest of the prompts will be the specified features of the image.

Sample Code:

```
"model": "gpt-4-vision-preview",
    "messages": [
        {
            "role": "user",
            "content": [
                {"type": "text", "text": f"Make a story based on the image
features?"},
                {"type": "text", "text": f"Landmark information from the
image in the order: ( name of place, street address #, street address,
neighborhood, city, district, region, country, postal code): +
{detect_landmarks(image)}"},
                {"type": "text", "text": f"The image contains these objects
and descriptions: +  {multiple_object(image)}"},
                {"type": "text", "text": f"Text features in the image:
+  {detect_text(image)}"},
                {"type": "text", "text": f"Logos in the image:
+  {detect_logo(image)}"},
                {"type": "text", "text": f"Colors in the image: +
{detect_properties(image)}"} if detect_properties(image) != "Hard time
deciphering color" else {"type": "text", "text": "Hard time deciphering
color"},
                ],
        }
```

F. Essential side implementations

The geocoder package in Python outputs more information about a landmark given its latitude and longitude coordinates. Since the Google Vision API landmark detection outputs the coordinates, we can pass them through the geocoder package and receive even more information about the landmark to implement into the story-telling script

Figure 6: sample photo from the internet of the city of Rochester

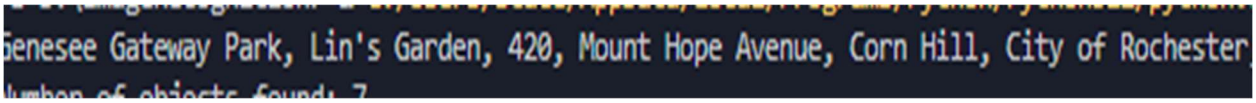

Figure 7: The output of passing Figure 5 into the landmark detection function

Using geocoder package and extracting the coordinates from Figure 6's image, the geocoder processed the coordinates to output the results in Figure 7. Figure 7 displays the name of, the address, city, neighborhood district, zip code, state, and country.

IV. Results & Display

After several tweaks to the script, it was successful in producing a compelling story based on the image. Users were able to pass an image URL and the script was able to extract features from the image and prompt the GPT-4 model to produce a story from the image. Despite the success, I felt the GPT-4 model can do more than produce a story- it can respond to questions about the story. Given the massive training dataset, OpenAI makes their GPT models trained on, responding to users' questions based on a story should be very doable. Therefore, I added a

reoccurring function where users can ask as many questions as they want about the story and the model will respond as adequately as it can.

The script was written in Visual Studio Code, a prominent IDE made by Microsoft for various programming languages. To run the script, users must select 'Run Python File' on the top right of the screen, and the output is displayed in the terminal. However, for a better user-interactive experience, I made a GUI display using the Tkinter package to have a graphical display for better user experience.

Figure 8: GUI Display using the Tkinter package for Thesis

Note: We will reuse -Figure 7 and pass its URL to the script.

Figure 8 is the GUI display I made in Python using the Tkinter package. The figure provides an input box for the image URL and another box to ask questions about the story once produced. Users can pass an image URL into the input box and select submit which prompts the GPT model to print the story on the interface. Once printed, the question box will invoke functionality for users to ask questions about the story.

Output:

```
Once upon a time in the bustling neighborhood of Corn Hill in the city of
Rochester, there was a beautiful park called Genesee Gateway Park. It was a popular
spot for locals to come and relax, take a stroll, or have a picnic with their loved
ones.

One day, a young couple decided to meet at the park for a romantic date. They had
planned to have lunch at Lin's Garden, a Chinese restaurant located at 420 Mount Hope
Avenue, just a short walk from the park. The couple had heard great things about the
restaurant's authentic cuisine and cozy atmosphere, and they were excited to try it
out.

As they strolled hand in hand through the park, they admired the lush greenery and
peaceful surroundings. The park was a welcome escape from the hustle and bustle of the
city, and they felt grateful to have such a beautiful spot in their neighborhood.

When they arrived at Lin's Garden, they were not disappointed. The food was
delicious, and the service was friendly and attentive. They enjoyed a leisurely meal,
savoring each dish and each other's company.

After lunch, they walked back to the park, feeling content and happy. They found a
quiet spot under a shady tree and lay down on a blanket to relax. As they looked up at
the blue sky and listened to the birds singing, they knew that Genesee Gateway Park
and Lin's Garden would become their new favorite spot in the City of Rochester.
```

V. Discussion

A. Successes

The results of the script have turned out to be successful. It completes the job of making a story and answers denotative questions about the story. In addition, if a user re-enters the same image URL, a different story is produced each time which provides various scenarios. After testing various images, the script proved to produce a compelling story based on the visual features.

B. Drawbacks

The significant drawback in executing this script would be the restraints by the OpenAI API. Given the API is not free, users have a token limit which outputs errors if there are not sufficient funds in the account. There have been errors where tentatively it could have been an internal error with the code but it turned out to be just a lack of funds in the OpenAI account. In addition, the Google Vision API was not always able to detect landmarks in images despite the visibility which can lead to inaccuracies in stories when users ask questions such as, “Where does this story take place?” Another major drawback in working on this thesis was the Google Vision APIs inability to capture all objects in an image. The API can capture emphasized objects but it misses minor background objects in an image which could have provided better details for the story.

C. Improvements

Fine-tuning is an essential method in machine learning where it assists pre-trained models in becoming more accurate. One of the inabilities of GPT-4’s model is answering connotative questions users ask. Whenever a user inputs a question that is rather emotional or personal, the model is unable to answer the question. However, through fine-tuning, it can add new data to pre-existing data to develop better accuracy and performance. This is an essential tool in deep learning to improve models and reduce inaccuracies.

D. Future Work

This thesis has the potential to be utilized in classrooms and be used for accessibility services. For many individuals, it is difficult to be creative or answer questions about images on

ELA exams or English tests. Through this thesis, people can develop a better understanding of stories based on images and ask questions to the point of comprehension. The full work of the thesis can be found on the GitHub link, https://github.com/sdo27/ImageNarration.

Overall, artificial intelligence has made significant advancements in the world. However, there is still potential to improve and utilize its capabilities. This story-telling script has performed a user-interactive experience in producing a story based on an image, but there are inaccuracies. However, inaccuracies can be fixed, and improvements can always be made to improve performance.

References

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. https://doi.org/10.1145/3442188.3445922

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., & Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877-1901.

Coursera. (2024). What is OpenAI? Everything you need to know. Retrieved May 23, 2024, from www.coursera.org/articles/what-is-openai

Google Cloud. (2023). Google Vision API. Retrieved from https://cloud.google.com/vision

Hariri, W. (2023). Unlocking the potential of ChatGPT: A comprehensive exploration of its applications, advantages, limitations, and future directions in natural language processing. arXiv preprint arXiv:2304.02017.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778. https://doi.org/10.1109/CVPR.2016.90

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097-1105.

Nantasenamat, C. (2023, August 10). Beginner's guide to OpenAI API. Medium, Data Professor.

Retrieved from https://medium.com/data-professor/beginners-guide-to-openai-api-a0420bc58ee5

OpenAI. (2023). OpenAI API. Retrieved from https://openai.com/api

Shidaganti, G., Sanjana, R., Shubeeksh, K., Raman, V. M., & Thakshith, V. (2023). ChatGPT: Information retrieval from image using robotic process automation and OCR. In 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1264-1270). https://doi.org/10.1109/ICICCS56964.2023.10005123

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Scribd. (2024). Python - the fastest growing programming language: Related papers. Retrieved May 23, 2024, from www.scribd.com/document/543296168/IRJET-V4I126620200101-109100-1nbuifw-with-cover-page-v2

Tian, Y. (2020). Artificial intelligence image recognition method based on convolutional neural network algorithm. IEEE Access, 8, 125731-125744. https://doi.org/10.1109/ACCESS.2020.3006097

Acknowledgments

Thank you to Computing Sciences thesis advisor Dr.Ning Yu for helping me on this journey of exploring the potential of OpenAI. The idea of image recognition was posed to him as well as implementing OpenAI into this thesis due to its endless capabilities.

Index

Acknowledgments, 4, 24
AI, 2, 10
APIs, 8, 12, 13, 14
artificial intelligence, 4
ChatGPT, 10, 22
Coursera, 11, 22
deep learning, 5
fine-tuning, 11
Google Vision API, 3, 7, 8, 9, 10
GPT, 6, 8, 11, 12, 17
image input, 7
Image Recognition, 6
Microsoft, 10, 17
natural language processing, 2, 5, 22
NLP, 5, 6, 7
object detection, 6, 9
OCR, 8, 14, 22
OpenAI, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15, 17, 22, 24
Python, 10, 11, 12, 13, 14, 17, 23
References, 4, 21
Shidaganti, 22
storytelling, 1
Table of Contents, 3
Text Generation, 6
Visual Studio Code, 17