

DESCRIBE AND GENERATE GAME USING GRADIO



By:

Name: Vaishnavi Patil

ID: 20133

Course: Generative AI-driven Intelligent Apps

Prof:

Name: Dr. Chang, Henry

Email: Henry.Chang@sfbu.edu



TABLE OF CONTENTS

- Introduction
- Design
- Implementation
- Testing
- Enhancement Ideas
- Conclusion
- References
- Appendix

INTRODUCTION



Objective: Create an interactive game that generates captions and images from user uploaded images and converts the captions into audio.



Technologies Used: Gradio, PIL, Requests, OpenAI API.



Significance: Enhances creativity and engagement through AI-generated content.

DESIGN

Why This Design?

- **Identify Problems:**

1. Limited user interaction with image generation.
2. Need for audio output to enhance accessibility.

- **Investigate Solutions:**

1. Explore AI models for image captioning, image generation, and TTS (Text-to-Speech).

- **Theoretical Comparison:**

Solution 1: Using separate APIs for each function.

Solution 2: Integrated solution using a unified API for efficiency.

IMPLEMENTATION



How It Was Done?



Components:

Image Upload: User uploads an image.

Caption Generation: API generates a caption based on the uploaded image.

Image Generation: New image created based on the caption.

Audio Generation: Caption converted to audio and saved.



Technologies:

Gradio for UI,
PIL for image handling
Requests for API calls.

TESTING

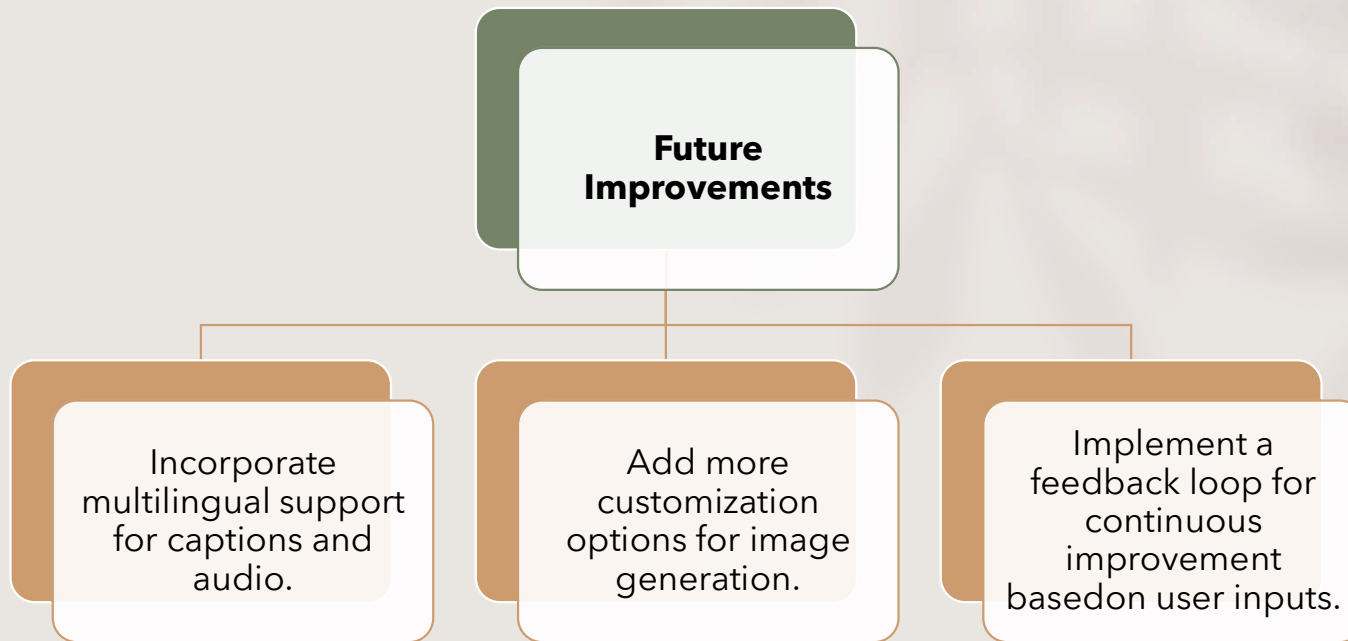
Testing Methodology:

- ✓ Unit tests for each function (image generation, captioning, audio).
- ✓ User acceptance testing (UAT) for overall experience.

Results:

- ✓ Successful image generation and caption accuracy.
- ✓ Smooth audio playback without latency.

ENHANCEMENT IDEAS



CONCLUSION



Summary:



Developed an innovative game that bridges text, audio, and visual creativity.



Demonstrated the effectiveness of combining multiple AI models for enhanced user experience.



Impact:



Engaged users through a creative platform that combines various forms of media.

REFERENCES

Key References:

AI Model Documentation (e.g., OpenAI)

Gradio Documentation

HuggingFace Documentation

Research articles on image captioning and audio synthesis.

Appendix

```
17 # Function to generate an image from a text prompt
18 def generate_image(prompt):
19     headers = {
20         "Authorization": f"Bearer {hf_api_key}"
21     }
22     payload = {
23         "inputs": prompt,
24         "parameters": {
25             "guidance_scale": 7.5,
26             "num_inference_steps": 50
27         }
28     }
29
30     response = requests.post(TTI_ENDPOINT, headers=headers, json=payload) # type: ignore
31
32     # Process response to display image
33     if response.status_code == 200:
34         try:
35             image_data = response.content
36             image = Image.open(io.BytesIO(image_data))
37             return image
38         except Exception as e:
39             print("Error processing image data:", e)
40             raise
41     else:
42         print(f"Error {response.status_code}: {response.text}")
43         raise Exception(f"Request failed with status code {response.status_code}. Details: {response.text}")
44
45 # Function to generate a caption from an uploaded image
46 def generate_caption(image):
47     headers = {
48         "Authorization": f"Bearer {hf_api_key}"
49     }
50     # Convert image to base64
51     buffered = io.BytesIO()
52     image.save(buffered, format="PNG")
53     image_base64 = base64.b64encode(buffered.getvalue()).decode("utf-8")
54
55     payload = {
56         "inputs": image_base64
57     }
58
59     response = requests.post(ITT_ENDPOINT, headers=headers, json=payload) # type: ignore
60
61     # Extract and return caption from response
62     if response.status_code == 200:
63         result = response.json()
64         if isinstance(result, list) and len(result) > 0 and 'generated_text' in result[0]:
65             return result[0]['generated_text']
66         else:
67             raise ValueError("Unexpected response structure from image captioning API")
68     else:
69         print(f"Error {response.status_code}: {response.text}")
70         raise Exception(f"Request failed with status code {response.status_code}. Details: {response.text}")
```

GenerateGameUsingGradio.py X

GenerateGameUsingGradio.py > generate_audio

```
72 # Function to generate audio from a text caption
73 def generate_audio(text):
74     headers = {
75         "Authorization": f"Bearer {hf_api_key}"
76     }
77     payload = {
78         "inputs": text
79     }
80
81     response = requests.post(TTS_ENDPOINT, headers=headers, json=payload) # type: ignore
82
83     # Process audio response and save to a temporary file
84     if response.status_code == 200:
85         audio_data = response.content
86         # Save the audio to a specific file path
87         output_audio_file = "Audios/generated_audio.wav"
88         with open(output_audio_file, "wb") as f:
89             f.write(audio_data)
90         return output_audio_file # Return the file path
91     else:
92         print(f"Error {response.status_code}: {response.text}")
93         raise Exception(f"Request failed with status code {response.status_code}. Details: {response.text}")
94
95 # Combined function for generating caption, image, and audio
96 def caption_and_generate(image):
97     caption = generate_caption(image)
98     generated_image = generate_image(caption)
99     audio_output = generate_audio(caption)
100     return [caption, generated_image, audio_output]
101
102 # Gradio interface for user interaction
103 with gr.Blocks() as demo:
104     gr.Markdown("Describe and Generate game")
105
106     # UI elements: Image input, button, and output fields
107     image_input = gr.Image(label="Upload an Image", type="pil")
108     btn_all = gr.Button("Generate Caption, Image, and Audio")
109     caption_output = gr.Textbox(label="Generated Caption")
110     image_output = gr.Image(label="Generated Image")
111     audio_output = gr.Audio(label="Generated Audio")
112
113     # Define button action
114     btn_all.click(fn=caption_and_generate, inputs=[image_input], outputs=[caption_output, image_output, audio_output])
115
```

127.0.0.1:7860

67%



Describe and Generate game

Upload an Image



Generate Caption, Image, and Audio

Generated Caption

Generated Image



127.0.0.1:7860

67%



Generate Caption, Image, and Audio

Generated Caption

a group of people sitting around a table with laptops

Generated Image



Generated Audio



0:00



1x



0:03

URLS

Google Slides:

https://docs.google.com/presentation/d/1buh9EDI13CIHrcED3Vn72U4rgEcl-Q_-JorrExcaLqs/edit?usp=sharing

GitHub Link:

<https://github.com/vaishnavi477/Machine-Learning/tree/main/Describe%20and%20Generate%20Game%20Using%20Gradio>



THANK YOU