

Simon Tesfatsion

GenAI Game on Gradio

[GitHub Link](#)

[Google Slides Link](#)

Introduction

- An interactive web application built with Gradio
- Combines image captioning and image generation AI models
- Creates an engaging game-like experience for users
- Features a clean, single-window interface

Technology Stack

Core Technologies:

- Gradio: Frontend framework for ML applications
- Python: Backend implementation
 - PIL (Python Imaging Library): Image processing
 - Pre-trained AI Models: Vision-Language Model for captioning(`blip-image-captioning-base`)
 - Diffusion Model for image generation(`stable-diffusion-3.5-large`)

Image Captioning Implementation

How It Works (sample Structure):

```
def captioner(image):  
    # Convert PIL image to model-compatible format  
    preprocessed_image = preprocess_image(image)  
  
    # Generate caption using vision-language model  
    caption = vision_model.generate_caption(preprocessed_image)  
  
    return caption
```

Full code implementation available on github: <https://github.com/Montegan/gradiolImage>

Image Captioning Implementation

Key Features:

- Processes images of any size to 200x200
- Generates descriptive captions in natural language
- Optimized for real-time performance

Image Generation Implementation:

How It Works (sample Structure):

```
def generate(caption):  
    # Preprocess caption for the model  
    processed_prompt = preprocess_text(caption)  
  
    # Generate image using diffusion model  
    generated_image = diffusion_model.generate(  
        prompt=processed_prompt,  
        image_size=(200, 200)  
    )  
  
    return generated_image
```

Full code implementation available on github: <https://github.com/Montegan/gradiolImage>

Gradio Interface Design:

How It Works (sample Structure):

```
with gr.Blocks() as demo:
    with gr.Row():
        with gr.Column():
            image_upload = gr.Image(
                label="Your first image",
                type="pil",
                width=200,
                height=200
            )
            btn_caption = gr.Button("Generate Caption")

        with gr.Column():
            image_output = gr.Image(
                label="Generated Image",
                width=200,
                height=200
            )
            btn_image = gr.Button("Generate Image")

    caption = gr.Textbox(label="Generated Caption")
```

Full code implementation available on github: <https://github.com/Montegan/gradioImage>

Gradio Interface Design:

Design Choices:



- Two-column layout for visual flow
- Consistent image dimensions
- Intuitive user interaction path

User Flow

- User uploads an image (200x200)
- System generates caption using Vision-Language model
- Caption displayed in textbox
- System generates new image based on caption
- New image displayed (200x200)

Demo:

Describe-and-Generate Game



Generate Caption

Generated Caption

a small dog and a cat playing in the grass

Generate Image

Use via API · Built with Gradio

9:01 PM 10/30/2024

Project Reference Materials

GitHub Link :

<https://github.com/Montegan/gradiolImage>

Google Slides Link :

https://docs.google.com/presentation/d/11cjat3jfK_sJ84Oy-sIILJhBqA2TibBrRRn50nOv2pE/edit?usp=sharing

“

Thank You

Simon Tesfatsion