Simon Tesfatsion

GenAl 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 Al 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/gradiolmage

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):
    processed prompt = preprocess text(caption)
    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/gradioImage

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,
            btn caption = gr.Button("Generate Caption")
        with gr.Column():
            image output = gr.Image(
                label="Generated Image",
                width=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:

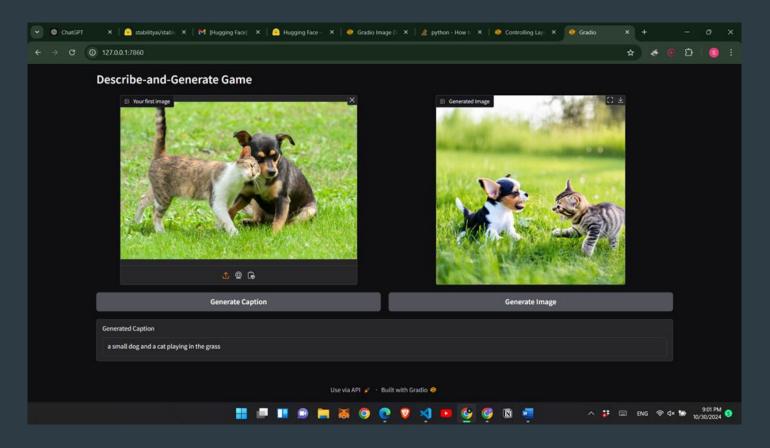
Design Choices:

- 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:



Project Reference Materials

GitHub Link:

https://github.com/Montegan/gradioImage

Google Slides Link:

https://docs.google.com/presentation/d/11cjat3jfK_sJ840y-sIILJhBqA2TibBrRRn50n0v2pE/edit?usp=sharing



Thank You

Simon Tesfatsion