Exploring Prompting Techniques for AI Video Generation

Aim

The aim of this experiment is to explore various prompting techniques for generating videos using AI models. The focus is on understanding how different prompt structures—such as simple versus detailed prompts—affect the quality, coherence, and style of the generated videos.

Softwares Required

1. **Python (3.8+)**
2. **IDE:** Jupyter Notebook or VS Code
3. **Python Libraries:** requests (for API calls), cv2 (optional, for handling video data)
4. **Video Generation AI Tools:** Runway ML’s Gen-2, Pika Labs (for experimentation in video generation)
5. **API Keys:** Required for Runway ML or other relevant AI models.

Experiment Design

1. **Prompt Variations:** Create three different prompt structures to observe their effects on video generation:
    - **Simple Prompt:** Basic description, e.g., "A forest with sunlight filtering through trees."
    - **Detailed Prompt:** Includes specific visual and contextual elements, e.g., "A serene, dense forest with sunlight filtering through tall, green trees, casting shadows on a narrow path covered in fallen leaves."
    - **Contextual/Story-Based Prompt:** Adds movement, mood, and details, e.g., "A slow, cinematic pan through a tranquil forest as sunlight filters through the trees. Leaves rustle gently in a soft breeze, with golden sunlight casting dappled shadows."
2. **Video Generation and Analysis:** Use the above prompts in an AI video generation model to observe and compare the generated videos.

Code

```
import requests

# API Key for Runway ML
API_KEY = "your_runway_api_key" url =
"https://api.runwayml.com/v1/models/gen-2/generate"  #
```

```
Function to generate video from a prompt def
generate_video(prompt):
    headers = {"Authorization": f"Bearer {API_KEY}"}
    payload = {
        "prompt": prompt,
        "format": "mp4", # Specify video format
        "fps": 24,
        "duration": 5 # Duration in seconds
    } response = requests.post(url, headers=headers,
    json=payload) if response.status_code == 200:
        video_url = response.json().get("video_url")
        print("Generated Video URL:", video_url)
        return video_url
    else: print("Video generation error:",
        response.json())

# Define Prompts simple_prompt = "A forest with sunlight filtering through trees."
detailed_prompt = "A serene, dense forest with sunlight filtering through tall, green trees, casting
shadows on a narrow path covered in fallen leaves." contextual_prompt = "A slow, cinematic pan
through a tranquil forest as sunlight filters through the trees. Leaves rustle gently in a soft breeze,
with golden sunlight casting dappled shadows."

# Generate Videos simple_video_url =
generate_video(simple_prompt) detailed_video_url =
generate_video(detailed_prompt) contextual_video_url =
generate_video(contextual_prompt)
```

Output and Result

Each prompt generates a video based on its description. Compare the outputs based on:

1. **Quality:** Video resolution and visual details.
2. **Coherence:** Smoothness and logical sequence of frames.
3. **Style and Mood:** How well the video reflects the intended mood or atmosphere.

Results Analysis:

- **Simple Prompt:** Often yields a generic video with basic elements but may lack intricate details.
- **Detailed Prompt:** Typically produces a richer, more visually engaging video with defined elements.
- **Contextual Prompt:** Adds storytelling aspects, resulting in videos with enhanced mood and coherence.