Text-to-Image Generator with Consistory

RUDRESH GOWDA S, SRUJAN K S ,Mrs. ROOPA C M
Student ,Dept.of CSE, Government polytechnic, Harihara, Karnataka, India
Student ,Dept.of CSE, Government polytechnic, Harihara , Karnataka, India
Senior scale Lecturer,Dept.of CSE,Government polytechnic, Harihara, Karnataka, India

1. Abstract

Artificial Intelligence (AI) and computer vision technologies have evolved rapidly, enabling new creative possibilities. One of the most exciting developments is text-to-image generation, where a system creates realistic images based on a natural language description. However, traditional models often struggle with subject consistency, making it hard to generate images that represent the same character across different scenes.

Our project, "Text-to-Image Generator with Consistory," addresses this problem using NVIDIA's Consistency Model, known as Consistory. This model enables consistent subject generation across multiple scenes without needing retraining. We developed a web application using Streamlit for the frontend and integrated NVIDIA's Consistency Model API as the backend.

The system allows users to input a subject description and two different scene prompts. It generates high-quality, visually coherent images, maintaining subject identity across scenes. The application is useful for storytelling, branding, education, and entertainment.

2. Problem Statement

Text-to-image models like DALL·E 2 and Stable Diffusion have advanced creative workflows but suffer from a critical weakness: they fail to preserve subject identity across multiple generated images. This poses a major challenge in industries like media, education, marketing, and design.

Consider a use case where an artist wants to illustrate the same character in different environments. Traditional models might generate different appearances for the same character, breaking continuity and realism. Manual correction through editing or retraining models is time-consuming and technically complex.

The main issues are:

Inconsistent subject portrayal across scenes.
Need for fine-tuning or manual editing.
Complex technical processes unsuitable for non-experts.

Our goal was to create a simple, accessible system that uses the NVIDIA Consistency Model to solve these problems. It should allow users to input simple prompts and generate consistent images automatically, reducing human effort and enhancing creative workflows.

3. Methodology

Planning and Design

The project began with extensive research on text-to-image models, understanding limitations, and studying NVIDIA's consistency approach. Our system was designed to:

- Provide an easy-to-use interface.
- Integrate the Consistency Model API.
- Deliver real-time results.

Frontend Development

Using Streamlit, we developed a lightweight and dynamic web interface. Key features include:

- Prompt input boxes for subject and two scenes.
- Generate button to trigger image creation.
- Image display and download options.

Streamlit allowed us to build a responsive interface quickly, with real-time feedback and easy deployment.

Backend Development

We used Python's "requests" library to send JSON-based API requests to the NVIDIA server. The steps included:

- Collecting subject and scene prompts.
- Formatting the request payload.
- Sending the POST request with authentication.
- Receiving and decoding base64-encoded image data.
- Displaying images in the frontend.
- Subject Tokenization

Subject tokens are keywords that anchor important visual features, like "woman, red dress, glasses." Tokens are essential for preserving identity and were extracted automatically from subject prompts.

Error Handling

We implemented robust error handling using try-except blocks. The app gracefully manages:

- **Missing API keys.**
- **Network failures.**
- **API response errors.**
- **User-friendly error messages and loading indicators improve the experience.**

Testing and Optimization

We conducted iterative testing with different prompts and scenes. Fine-tuning parameters like CFG scale and noise control helped optimize outputs for maximum subject consistency.

4.Algorithms and Flowchart

Algorithms

Input Algorithm:

1. User inputs subject and scene prompts.
2. Parse subject tokens from the subject prompt.
3. Prepare the API request payload.

API Communication Algorithm:

1. Send POST request to the Consistency Model API.
2. Await and validate the response.
3. Handle errors if the request fails.

Image Decoding Algorithm:

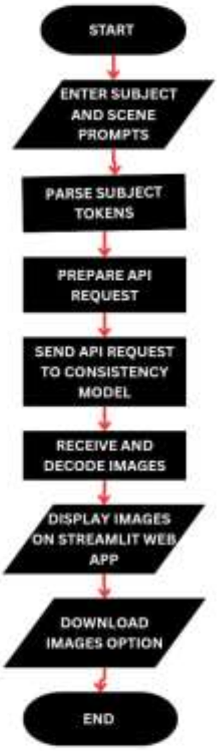
1. Extract base64 data from the API response.
2. Decode images.
3. Display images and enable download.

Error Handling Algorithm:

1. Catch HTTP and network errors.
2. Display clear messages to the user.
3. Allow retry or guide user to correct inputs.

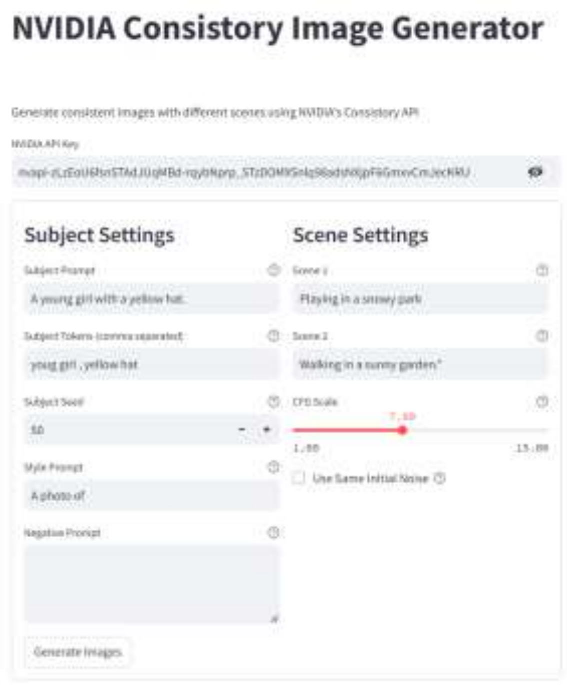
Flowchart

1. Start
2. User inputs subject and scenes
3. Parse subject tokens
4. Formulate API request
5. Send request to NVIDIA Consistency API
6. If success: decode and display images
7. If failure: show error and retry option
8. Allow user to download images
9. End


START
ENTER SUBJECT AND SCENE PROMPTS
PARSE SUBJECT TOKENS
PREPARE API REQUEST
SEND API REQUEST TO CONSISTENCY MODEL
RECEIVE AND DECODE IMAGES
DISPLAY IMAGES ON STREAMLIT WEB APP
DOWNLOAD IMAGES OPTION
END

5. Results and Achievements

Testing and Sample Cases



NVIDIA Consistory Image Generator

Generate consistent images with different scenes using NVIDIA's Consistory API

NVIDIA API Key: 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

Subject Settings

Subject Prompt: A young girl with a yellow hat.

Subject Tokens (comma separated): youg girl , yellow hat

Subject Seed: 50

Style Prompt: A photo of

Negative Prompt:

Scene Settings

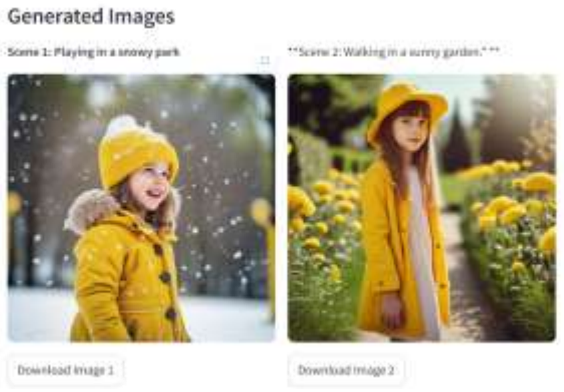
Scene 1: Playing in a snowy park

Scene 2: Walking in a sunny garden."

CFG Scale: 7.50 (1.00 – 15.00)

☐ Use Same Initial Noise

Case 1:
Subject: "A young girl with a yellow hat."
Scene 1: "Playing in a snowy park."
Scene 2: "Walking in a sunny garden."
Result: The girl's facial features, clothing, and accessories remained identical across both scenes.

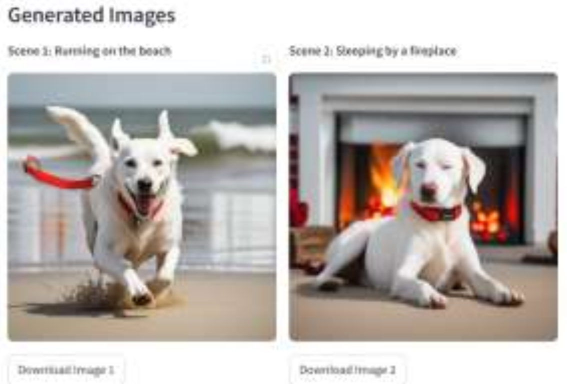


Generated Images

Scene 1: Playing in a snowy park

**Scene 2: Walking in a sunny garden." **

Case 2:
Subject: "A white dog with a red collar."
Scene 1: "Running on the beach."
Scene 2: "Sleeping by a fireplace."
Result: The dog's color, breed, and collar remained consistent, with accurate environmental adaptation.

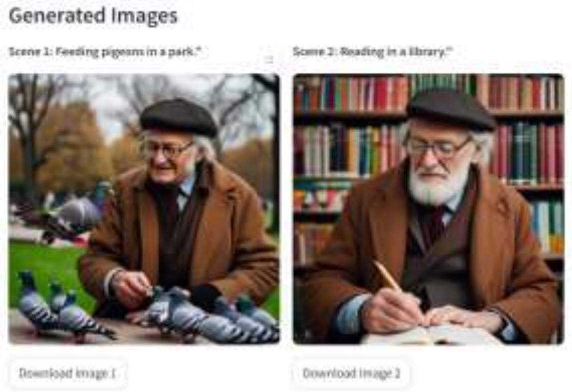


Generated Images

Scene 1: Running on the beach

Scene 2: Sleeping by a fireplace

Case 3:
Subject: "An old man wearing glasses and a brown coat."
Scene 1: "Feeding pigeons in a park."
Scene 2: "Reading in a library."
Result: Subject appearance was stable across scenes, validating model performance.



Generated Images

Scene 1: Feeding pigeons in a park."

Scene 2: Reading in a library."

Achievements

- Maintained high subject consistency across different prompts.
- Delivered images within 15-40 seconds.
- Enabled real-time creative exploration.
- Provided an intuitive, no-code solution.
- Avoided manual retraining processes.
- Successfully handled common API errors and failures.

Observations

Prompt structure strongly influenced results. Carefully defining subject tokens improved consistency. Variations in scene prompts were handled smoothly, though extreme changes sometimes caused minor identity drift, a limitation typical even in advanced models.

6. Conclusion

The "Text-to-Image Generator with Consistory" project successfully enables generating consistent images from text prompts using NVIDIA's Consistency Model. The system ensures subject identity across different scenes without retraining, making it useful for creative, educational, and branding purposes.

While minor limitations exist, the application demonstrates how AI can simplify digital content creation. Future improvements could add multi-subject support and offline usage, expanding its potential further.

7. References

1. **NVIDIA Research**. "Consistency Models for Text-to-Image Generation."
https://research.nvidia.com/labs/toronto-ai/consistency-models
2. **Streamlit Documentation**. "Build Data Apps with Streamlit."
https://docs.streamlit.io/
3. **Python Requests Library**. "Requests: HTTP for Humans."
https://requests.readthedocs.io/en/latest/