

Day43_Project_Image_to_Text_Using_Generative_AI

July 12, 2025

Image to Text Generation Using Google Generative AI (Gemini)

This project demonstrates how to build a simple image-to-text system using Google's Generative AI (Gemini 2.5 Flash model). We connect to Gemini using Python, send an image input, and receive an AI-generated description of the image.

What You'll Learn

- What is Generative AI and how it works
- How to use google-generativeai Python SDK
- How to send image input and receive meaningful responses
- How to display and format AI outputs using Markdown

Technologies Used

- Python
- Google Generative AI (google-generativeai)
- Gemini 2.5 Flash + Thinking models
- PIL for image input
- IPython for Markdown output

1 Install the Required Library

You only need to do this once.

```
!pip install -U google-generativeai
```

This command installs the google-generativeai library that allows us to use Google Gemini models directly in Python.

2 Import All Required Libraries

- google.generativeai → to interact with Gemini models
- PIL.Image → to load and display the image
- textwrap + Markdown → to display formatted model outputs

```
[1]: import google.generativeai as genai
import PIL.Image
import textwrap
from IPython.display import Markdown
```

3 Configure the API Key

- Before running, get your API key from

<https://makersuite.google.com/app/apikey>

Make sure:

- You've enabled the Generative Language API in your Google Cloud account.
- Don't share your key publicly.

Replace "Your key" with your actual key

```
[2]: genai.configure(api_key="Your key")
```

4 Define Markdown Output Formatter

This helps make the AI's response easier to read.

```
[3]: def to_markdown(text):
    text = text.replace('-', '•') # Optional: change bullets
    return Markdown(textwrap.indent(text, '> ', predicate=lambda _: True))
```

5 Load the Image

Make sure the image path is correct on your PC.

```
[5]: # This loads and shows your image.
img_path = r"C:\Users\Lenovo\Downloads\1.png" # Change this path if needed
img = PIL.Image.open(img_path)
img
```

[5]:



6 Generate Text from Image using Gemini 2.5 Flash Model

- `generate_content(img)` sends the image to Gemini
- Gemini returns a description or analysis of what's in the image

```
[6]: model = genai.GenerativeModel('models/gemini-2.5-flash-preview-05-20') # image_
      ↳ understanding model
      response = model.generate_content(img)
      to_markdown(response.text)
```

[6]: This image is a composite of various Hindu deities and one prominent saint. Here is a list of the figures depicted:

1. **Lord Shiva, Goddess Parvati, and Lord Ganesha (Shiva Parivar):** Central figures in the image. Shiva is blue •skinned with a trident, Parvati is in red and green holding baby Ganesha (elephant •headed deity).
2. **Goddess Lakshmi:** Top left, seated on a pink lotus, holding lotuses, symbolizing wealth and prosperity.
3. **Lord Brahma:** Top right, multi •headed (usually four, but three visible here), seated on a lotus, symbolizing creation.
4. **Goddess Durga / Mahishasuramardini:** Mid •left, multi •armed, riding a tiger, holding various weapons, symbolizing divine feminine power and protection.
5. **Lord Vishnu / Venkateswara (Tirupati Balaji):** Top right, a black stone idol with a tall crown, typically worshipped in Tirumala.
6. **Sai Baba of Shirdi:** Mid •right, an Indian spiritual master and fakir, revered by both Hindu and Muslim devotees.
7. **Lord Krishna:** Bottom left, blue •skinned, holding a flute, symbolizing divine love and joy.
8. **Lord Rama:** Bottom middle •left, holding a bow and arrow, symbolizing righteousness and duty.
9. **Lord Kartikeya / Murugan / Subramanya:** Bottom middle, golden •skinned, holding a vel (spear), a popular deity in South India.
10. **Lord Hanuman:** Bottom right, the monkey god, depicted in a meditative posture, symbolizing strength, devotion, and service.

7 Use Gemini 2.5 Thinking Model on the Same Image

This model is better for deep reasoning.

```
[7]: model = genai.GenerativeModel('models/gemini-2.5-flash-preview-04-17-thinking')
response = model.generate_content(img)
to_markdown(response.text)
```

[7]: Here is a description of the image:

The image is a vibrant composite featuring various Hindu deities and a saint arranged against a red background with a faint mandala pattern.

Prominent figures include:

1. **Center:** Shiva (blue •skinned with trident and snake), Parvati, and their son Ganesha (elephant •headed child), forming a family group.
2. **Top Center:** Brahma, depicted with multiple heads, seated on a lotus.
3. **Top Left:** Goddess Lakshmi, seated on a pink lotus, holding lotuses.
4. **Top Right:** Lord Venkateswara (a form of Vishnu), shown as a dark •skinned deity with a large crown and ornaments.
5. **Middle Left:** Goddess Durga (or a similar form), depicted with multiple arms, holding weapons, and riding a tiger.
6. **Middle Right:** Sai Baba of Shirdi, an Indian spiritual master, in his characteristic pose.
7. **Bottom Left:** Lord Krishna, depicted with blue skin, playing a flute, and wearing a peacock feather.

8. **Bottom Middle Left:** Lord Rama, holding a bow and arrow.
9. **Bottom Middle:** Lord Murugan (or Kartikeya), depicted in golden form, holding a spear.
10. **Bottom Right:** Hanuman, the monkey god, seated in meditation with a mace nearby. A small box (possibly containing sacred ash or prasad) is in front of him.

The overall composition brings together major deities from different branches of Hinduism and a revered saint, suggesting a representation of divine presence and blessings.

8 Conclusion

- We successfully built an image-to-text generation app using Gemini 2.5 Flash & Thinking models.
- Learned how to send image input to Gemini and interpret the response.
- Formatted the output using IPython for a clean notebook presentation.
- This project serves as a foundation for multi-modal AI applications, such as visual assistants, photo analyzers, and AI-powered document readers.