

Project Documentation:

TITLE:

AI Nutritionist App Using Gemini Pro: Your Comprehensive Guide to Healthy Eating and Well-being

Table of Contents

1. Introduction
2. Objective
3. System Overview
4. Technologies Used
5. Features
6. Architecture
7. Installation and Setup
8. Code Overview
9. User Guide
10. Future Enhancements
11. Conclusion

1. Introduction

The AI Nutritionist App is designed to provide users with personalised calorie assessments of meals by analysing food images. By leveraging Google Gemini's AI-powered model (Gemini 1.5 Flash) and Google

Generative AI, the app calculates the total calorie intake based on the food items identified from user-uploaded images. The app also allows users to provide textual input to guide the calorie estimation process.

2. Objective

The goal of the AI Nutritionist App is to streamline the process of calorie estimation for users who want to monitor their dietary intake. By analysing images of food and returning detailed calorie counts for each item, the app helps users make informed decisions about their meals and overall nutrition.

3. System Overview

The AI Nutritionist App comprises the following components:

- **User Input Interface:** Allows users to provide an image of their meal and a textual input prompt.
- **Google Gemini Integration:** Uses the Gemini 1.5 Flash model to generate the total calorie count by analysing food items in the image.
- **Calorie Estimation Output:** Displays the calories of each food item in the image and provides a structured response based on the input prompt.
- **Interactive UI:** Built using Streamlit, providing a smooth user experience for uploading images, receiving outputs, and interacting with the model.

4. Technologies Used

- **Streamlit:** Used to create the web-based user interface for the app. It allows users to upload images, input prompts, and view the calorie

analysis results.

- **Google Generative AI (Gemini):** Gemini 1.5 Flash is used to process both the image and input prompts and return accurate calorie estimates for food items.
- **Python-dotenv:** Used to load environment variables, such as the Google API key, from a ``.env`` file.
- **Pillow:** A Python Imaging Library used for image handling, including processing user-uploaded food images.
- **PyPDF2:** For potential future integration, used for working with PDF files if necessary.
- **Streamlit Extras:** To enhance the app's interface and functionality.

5. Features

- **Calorie Estimation:** Users can upload images of their meals, and the app provides calorie counts for each food item.
- **Text Prompt Input:** Users can provide additional instructions or prompts to customise the analysis (e.g., specifying ingredients or requesting specific analysis formats).
- **Image Upload:** The app accepts image files in ``.jpg``, ``.jpeg``, and ``.png`` formats and processes them to identify food items.
- **Interactive Display:** The uploaded image is displayed, and the calorie estimation results are shown in a clean, structured format.

6. Architecture

The architecture of the app is organized into three core layers:

- **Front-End (Streamlit):** Provides the user interface for uploading images and inputting prompts. Displays the estimated calories based

on the user's input.

- **Backend Processing (Python + Gemini Pro):** Processes the user's image using Google Generative AI's Gemini model to detect food items and calculate the corresponding calories.
- **Data Handling (Pillow and Streamlit File Uploader):** Handles the uploaded image and prepares it for analysis by the AI model.

7. Installation and Setup

Prerequisites

- Python 3.x
- Google API Key
- Streamlit account

Installation Steps

1. Install the Required Libraries:

Navigate to the project directory and install the dependencies from the `requirements.txt` file:

```
pip install -r requirements.txt
```

2. Set Up Environment Variables:

Create a `.env` file and add your Google API key:

```
GOOGLE_API_KEY=<your_google_api_key>
```

3. Run the Streamlit App:

After setting up, launch the app using the following command:

```
streamlit run app.py
```

8. Code Overview

The code for the app is structured to handle the following components:

1. Google Gemini API Integration:

```
genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))
```

This code configures the Generative AI model (Gemini) using the API key, which is stored in an environment variable.

2. Gemini Model Function:

```
def get_gemini_response(input, image, prompt):  
    model = genai.GenerativeModel('gemini-1.5-flash')  
    response = model.generate_content([input, image[0], prompt])  
    return response.text
```

This function processes the input prompt and image and returns the response text generated by the Gemini model, which contains calorie information for each food item in the image.

3. Image Processing:

```
def input_image_setup(uploaded_file):  
    if uploaded_file is not None:  
        bytes_data = uploaded_file.getvalue()  
        image_parts = [  
            {  
                "mime_type": uploaded_file.type,  
                "data": bytes_data  
            }  
        ]
```

```
        return image_parts
    else:
        raise FileNotFoundError("No file uploaded")
```

This function reads the uploaded image, converts it into bytes, and prepares it for sending to the Gemini model.

4. Streamlit Front-End:

```
input = st.text_input("Input Prompt: ", key="input")
uploaded_file = st.file_uploader("Choose an image...", type=["jpg", "jpeg",
"png"])
if uploaded_file is not None:
    image = Image.open(uploaded_file)
    st.image(image, caption="Uploaded Image.", use_column_width=True)
submit = st.button("Tell me the total calories")
if submit:
    image_data = input_image_setup(uploaded_file)
    response = get_gemini_response(input_prompt, image_data, input)
    st.subheader("The response is")
    st.write(response)
```

The code provides the user interface for text input, image uploading, and a button for submitting the request. It processes the image and input and then displays the calorie estimate.

9. User Guide

1. **Launch the app** by running the `streamlit run app.py` command.
2. **Input the prompt** related to the food analysis you want (e.g., "Analyze

the calories in this image").

3. **Upload an image** of the food you want analyzed by clicking the file uploader.

4. **Click the "Tell me the total calories" button** to submit the request.

5. **View the response**, which includes a calorie breakdown of the food items in the uploaded image.

10. Future Enhancements

- **Multi-User Support:** Extend the app to support multiple users with personalized calorie and nutrition tracking.
- **Nutrient Breakdown:** Provide detailed nutritional information such as fats, carbohydrates, and proteins, in addition to calories.
- **Recipe Recommendations:** Suggest healthy recipes based on the identified food items.
- **Calorie Tracker Integration:** Add functionality for users to log meals over time and track total calorie intake for the day or week.
- **Wearable Device Integration:** Sync the app with fitness devices to track calories burned and adjust meal plans accordingly.

11. Conclusion

The AI Nutritionist App leverages state-of-the-art AI and data analysis tools to deliver a personalized and interactive experience for monitoring calorie intake. By combining Google Gemini's AI capabilities with a user-friendly Streamlit interface, the app helps users stay informed about their nutrition and make healthier food choices effortlessly.