Nutri Guide – Image to Text Generation

Team Name

- NutriTech Innovators

Team Members

- Yamini Reddy Pesaru
- Priyanka Bolem
- Supriya Panyala Rajaiahgari

Abstract

NutriGuide is an AI-powered web application designed to analyze food images and provide valuable insights into nutrition, health benefits, and famous recipes. By leveraging **TensorFlow** for image classification, **Hugging Face models** for language processing, and **Streamlit** for the user interface, NutriGuide delivers a seamless experience for users who want to learn more about the food they consume.

1 Tools and Technologies

Category	Tools/Technologies Used
AI/ML	TensorFlow, Hugging Face
Frontend	Streamlit
Image Handling	OpenCV, PIL
NLP	LangChain, HuggingFace Hub, ChatHuggingFace
Cloud/API	HuggingFace Hub API
Languages	Python
DevOps	Git, GitHub
IDE	VS Code, Google Colab
Others	NumPy, Pandas, Base64, Requests

Figure 1: Tools and Technologies Used in the NutriGuide System

2 Methodology Diagram

User Uploads Image: The user provides a food image (e.g., burger.jpg) through the interface.

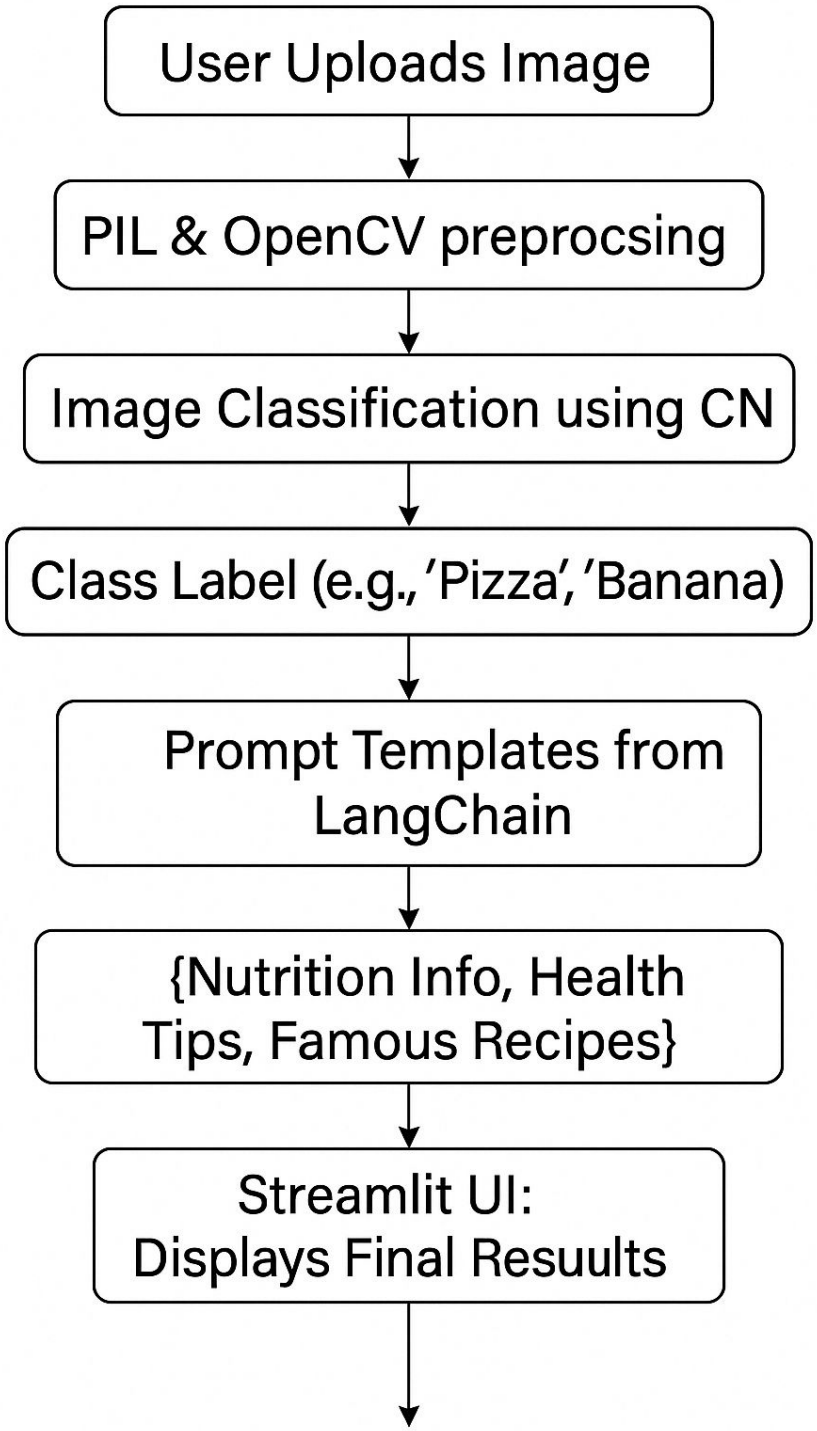

Figure 2: Block Diagram

PIL & OpenCV Preprocessing: The image is resized, normalized, and converted to the appropriate format using Python libraries PIL (Pillow) and OpenCV.

Image Classification using CNN: A Convolutional Neural Network (CNN) model, built using TensorFlow, classifies the food image into a category like 'Pizza' or 'Banana'.

Class Label: The model outputs a predicted food label (e.g., Pizza, Banana) based on the image input.

Prompt Templates from LangChain: The predicted label is passed into predefined prompt templates using LangChain to generate structured responses.

Output Categories: The system generates textual outputs based on the label including:

- Nutrition Info
- Health Tips
- Famous Recipes

Streamlit UI: The final results are displayed on the Streamlit web interface, showing prediction label, nutritional insights, health benefits, and recipe suggestions.

3 Detailed Implemantation

1. Install dependencies: TensorFlow, Streamlit, LangChain, HuggingFace Hub
2. Train or load a pre-trained food classification model
3. Design prompt templates for nutrition, benefits, and recipes
4. Connect the model and Hugging Face via LangChain
5. Set up the frontend in Streamlit to accept images
6. Display results with predictions and formatted model outputs
7. Test with multiple food items to validate outputs
8. Deploy using streamlit run or a tunnel like LocalTunnel/ngrok
9. Save the model (.keras), secure API keys using environment variables

4 Code Snippets

Installing all necessary packages

```
!pip install tensorflow==2.17.1
!pip install uvicorn==0.22.0
!pip install numpy==1.26.4
!pip install python-multipart
!pip install huggingface_hub
!pip install streamlit
!pip install langchain_huggingface
```

Image Preprocessing using PIL and OpenCV

```
import cv2
from PIL import Image
import numpy as np

def preprocess_image(image_path):
    img = Image.open(image_path).resize((224, 224))
    img = np.array(img) / 255.0
    return img
```

Prediction using TensorFlow Model

```
from tensorflow.keras.models import load_model

model = load_model('food_classifier.keras')
def predict_image(image_array):
    prediction = model.predict(image_array[np.newaxis, ...])
    return prediction.argmax()
```

Prompt Template with LangChain

```
from langchain.prompts import PromptTemplate

template = PromptTemplate.from_template(
    "Provide nutrition facts and one popular recipe for {food_item}."
)
response = template.format(food_item="Pizza")
```

References

1. **TensorFlow**: TensorFlow is an open-source machine learning framework. The specific version you are using can be found in the TensorFlow documentation.

   TensorFlow: An end-to-end open source machine learning platform. Retrieved from `https://www.tensorflow.org/`

2. **Streamlit**: Streamlit is a Python library used to build interactive web applications. It's used in your project to create the UI for food image classification.

   Streamlit - The fastest way to build data apps. Retrieved from `https://streamlit.io/`

3. **Hugging Face**: Hugging Face provides powerful models and APIs for text generation. In this project, it's used for generating nutrition information, health benefits, and recipes based on the predicted food item.

   Hugging Face Transformers. Retrieved from `https://huggingface.co/`

4. **LangChain**: LangChain is a framework designed for developing applications with large language models (LLMs). It is used in your project for creating prompts and chaining the outputs.

   LangChain: A framework for building LLM applications. Retrieved from `https://langchain.com/`

5. **PIL (Pillow)**: PIL (Pillow) is a Python Imaging Library used to process the uploaded food images for classification.

   Pillow - Python Imaging Library (PIL Fork). Retrieved from `https://pillow.readthedocs.io/en/stable/`

6. **Keras**: Keras is an open-source software library that provides a Python interface for neural networks, and it is used to load your trained food image classification model.

   Keras: The Python Deep Learning library. Retrieved from `https://keras.io/`

7. **Model Source**: **Microsoft Phi-3-mini-4k-instruct**: This is the specific model used in your project from Hugging Face, which provides detailed nutritional and health data generation based on food classification.

   Phi-3-mini-4k-instruct. Retrieved from `https://huggingface.co/microsoft/Phi-3-mini-4k-instruct`

GitHub Repository:

- Repository URL: `https://github.com/SupriyaRajaiahgari/GenAi`