Project Report: The Smartest AI Nutrition Assistant

Project Title

The Smartest AI Nutrition Assistant

Objective

The objective of this project is to design and implement an AI-powered nutrition assistant that can understand user inputs via text or image and provide accurate food identification along with nutritional information. It aims to assist users in making informed dietary choices through real-time analysis.

Problem Statement

In today's fast-paced world, many people struggle to track their nutrition accurately. Existing apps often require manual input and lack real-time recognition or intelligent suggestions. This project addresses the need for a smart, automated system that can recognize food from images and return reliable nutrition information.

Tools and Technologies Required

- Python 3.10+
- TensorFlow / Keras (MobileNetV2 for image classification)
- NumPy (image preprocessing)
- Requests (for API access)
- USDA FoodData Central API (for nutrition data)
- PIL / Keras image utilities (for image loading and manipulation)

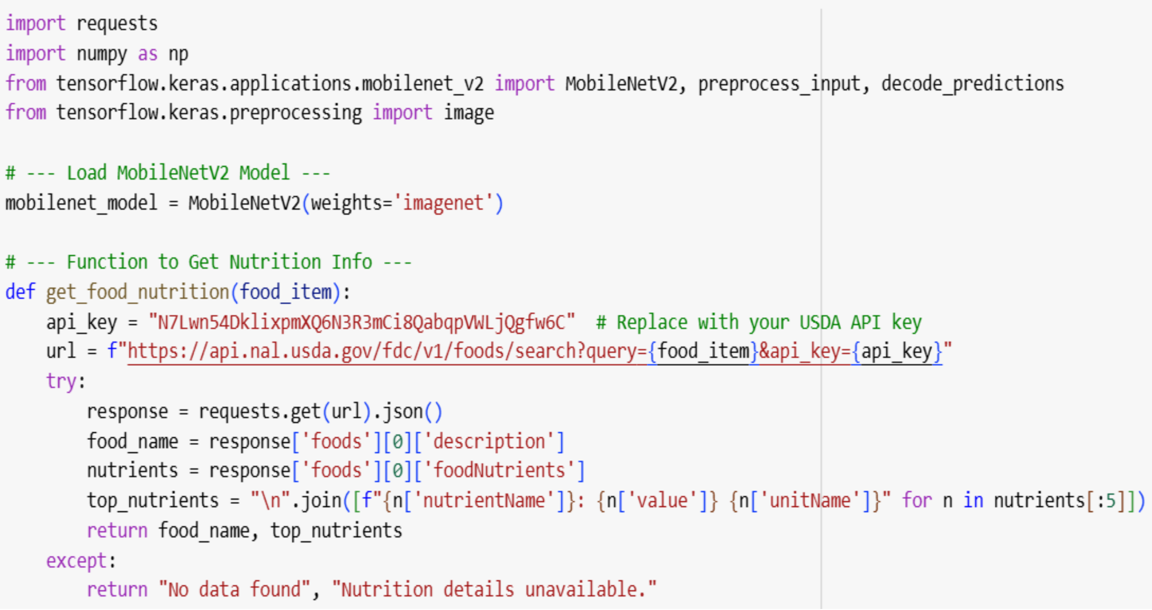
Methodology

The application operates in two modes: Text and Image. In text mode, users manually enter a food name, while in image mode, a food photo is analyzed using the MobileNetV2 model to predict the item. The food name is then used to fetch nutrition data from the USDA FoodData Central API. Top 3 predictions are shown for image input with a warning if model confidence is low.

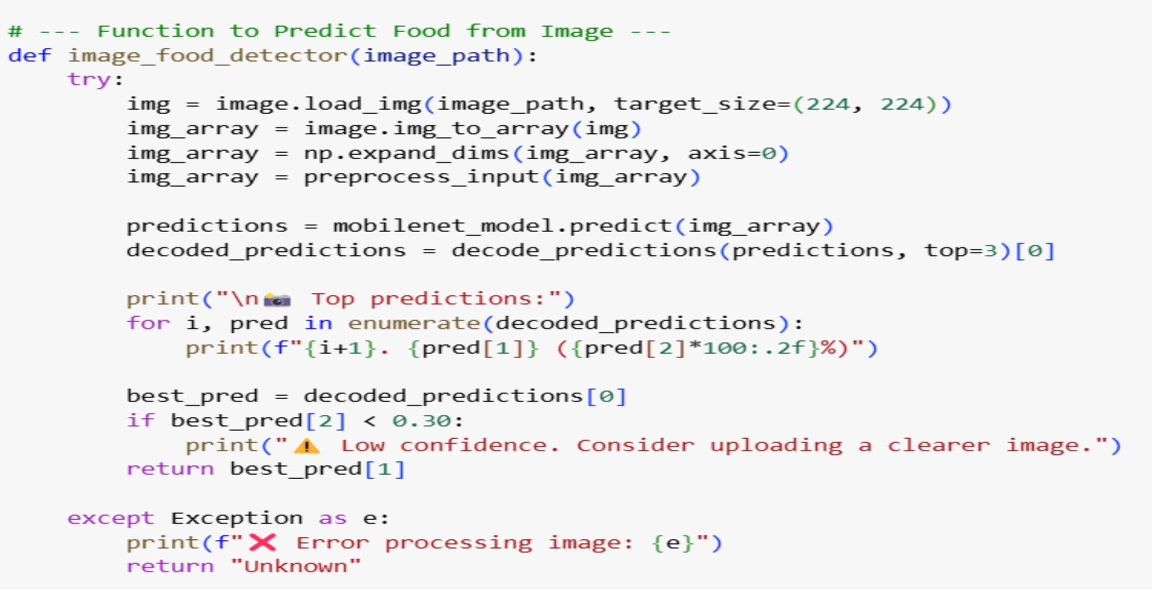
System Workflow

1. User selects input mode (text or image).
2. If image mode is selected, the image is loaded and passed through MobileNetV2.
3. Top food predictions are displayed with confidence scores.
4. The predicted (or entered) food name is sent to the USDA API.
5. Nutrient data is fetched and shown to the user.

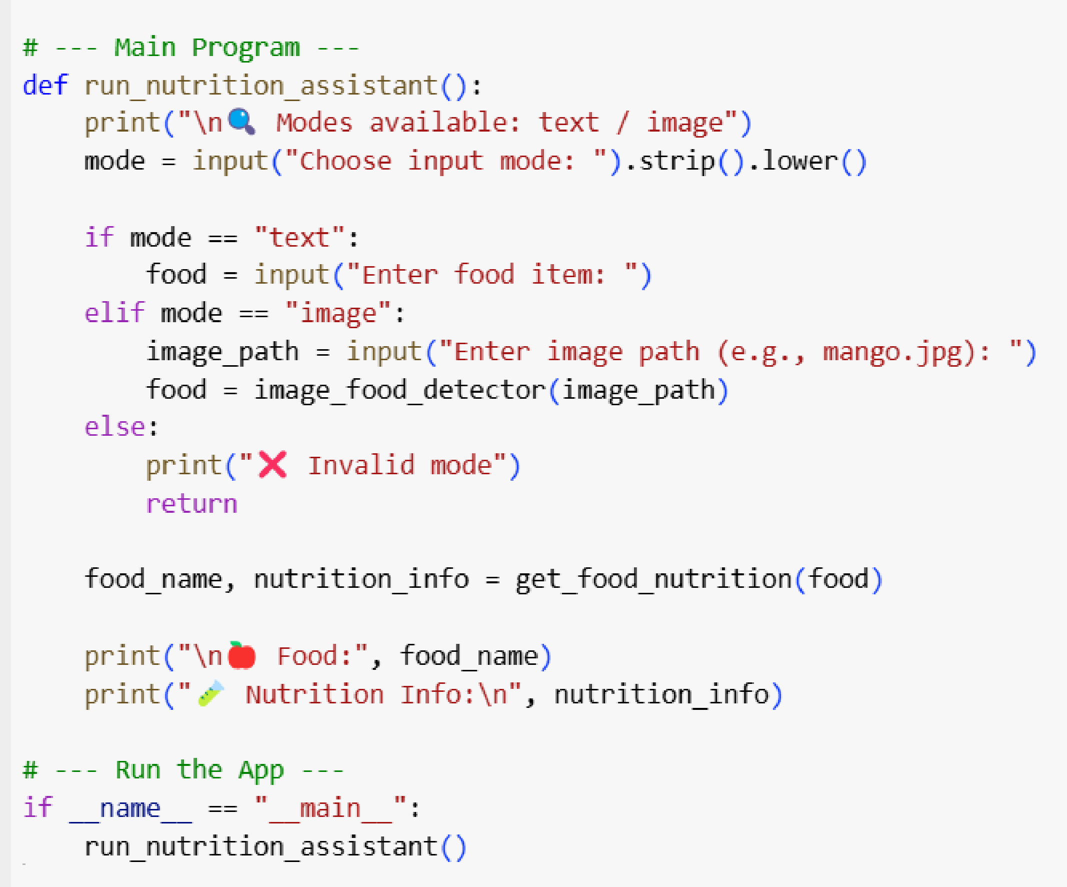
code&output



```python
import requests
import numpy as np
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2, preprocess_input, decode_predictions
from tensorflow.keras.preprocessing import image

# --- Load MobileNetV2 Model ---
mobilenet_model = MobileNetV2(weights='imagenet')

# --- Function to Get Nutrition Info ---
def get_food_nutrition(food_item):
    api_key = "N7Lwn54DklixpmXQ6N3R3mCi8QabqpVWLjQgfw6C"  # Replace with your USDA API key
    url = f"https://api.nal.usda.gov/fdc/v1/foods/search?query={food_item}&api_key={api_key}"
    try:
        response = requests.get(url).json()
        food_name = response['foods'][0]['description']
        nutrients = response['foods'][0]['foodNutrients']
        top_nutrients = "\n".join([f"{n['nutrientName']}: {n['value']} {n['unitName']}" for n in nutrients[:5]])
        return food_name, top_nutrients
    except:
        return "No data found", "Nutrition details unavailable."
```



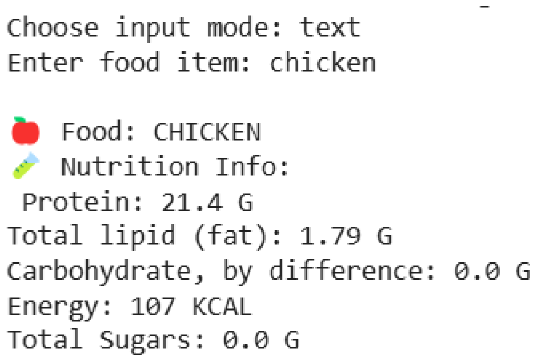
```python
# --- Function to Predict Food from Image ---
def image_food_detector(image_path):
    try:
        img = image.load_img(image_path, target_size=(224, 224))
        img_array = image.img_to_array(img)
        img_array = np.expand_dims(img_array, axis=0)
        img_array = preprocess_input(img_array)

        predictions = mobilenet_model.predict(img_array)
        decoded_predictions = decode_predictions(predictions, top=3)[0]

        print("\n📷 Top predictions:")
        for i, pred in enumerate(decoded_predictions):
            print(f"{i+1}. {pred[1]} ({pred[2]*100:.2f}%)")

        best_pred = decoded_predictions[0]
        if best_pred[2] < 0.30:
            print("⚠️ Low confidence. Consider uploading a clearer image.")
        return best_pred[1]

    except Exception as e:
        print(f"❌ Error processing image: {e}")
        return "Unknown"
```



```python
# --- Main Program ---
def run_nutrition_assistant():
    print("\n🔍 Modes available: text / image")
    mode = input("Choose input mode: ").strip().lower()

    if mode == "text":
        food = input("Enter food item: ")
    elif mode == "image":
        image_path = input("Enter image path (e.g., mango.jpg): ")
        food = image_food_detector(image_path)
    else:
        print("❌ Invalid mode")
        return

    food_name, nutrition_info = get_food_nutrition(food)

    print("\n🍎 Food:", food_name)
    print("🧪 Nutrition Info:\n", nutrition_info)

# --- Run the App ---
if __name__ == "__main__":
    run_nutrition_assistant()
```

input-Text(chicken)

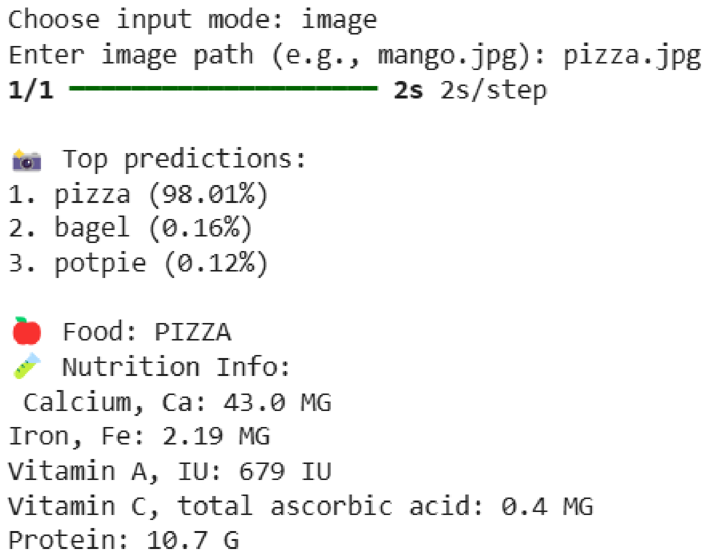
output:



```
Choose input mode: text
Enter food item: chicken

🍎 Food: CHICKEN
🧪 Nutrition Info:
 Protein: 21.4 G
Total lipid (fat): 1.79 G
Carbohydrate, by difference: 0.0 G
Energy: 107 KCAL
Total Sugars: 0.0 G
```

Input-image(pizza.jpg)

Output:



```
Choose input mode: image
Enter image path (e.g., mango.jpg): pizza.jpg
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step

📷 Top predictions:
1. pizza (98.01%)
2. bagel (0.16%)
3. potpie (0.12%)

🍎 Food: PIZZA
🧪 Nutrition Info:
 Calcium, Ca: 43.0 MG
Iron, Fe: 2.19 MG
Vitamin A, IU: 679 IU
Vitamin C, total ascorbic acid: 0.4 MG
Protein: 10.7 G
```

Future Scope

- Integrate voice input for hands-free interaction
- Use a custom-trained deep learning model for Indian and regional foods
- Deploy as a mobile/web app using Streamlit or Flask
- Include meal suggestions and diet plans based on user goals

Conclusion

This project successfully demonstrates how AI can be applied to simplify nutrition tracking. By using image recognition and real-time APIs, users receive instant and reliable nutritional information, paving the way for smarter dietary decisions.