7065 Group Project - SmartRecipe

Gong Fan
24439169@life.hkbu.edu.hk

Yang Sunchengrui
24478407@life.hkbu.edu.hk

Huangqiang Fang
24404616@life.hkbu.edu.hk

Abstract

SmartRecipe is an app that uses image recognition to identify food ingredients from photos. It addresses the challenge of meal planning and food waste by offering three key features: ingredient identification from user photos, personalized recipe recommendations, and nutritional analysis. The app helps users reduce food waste, eat healthier, and simplify meal preparation by suggesting what to cook with available ingredients. By combining these technologies, SmartRecipe creates a convenient solution for people with busy lifestyles who want to make better use of their food while making informed dietary choices.

1. Introduction

In today's fast-paced world, healthy eating and meal preparation can sometimes be a challenge, especially for those with busy lifestyles. SmartRecipe is an innovative app designed to solve this problem by using cutting-edge image recognition technology to identify ingredients directly from user-uploaded photos. This app goes beyond just identifying food—it helps users make smarter cooking choices by suggesting personalized recipes based on the available ingredients and providing detailed nutritional insights. The core idea behind SmartRecipe is to make meal preparation easier, reduce food waste, and empower users to make healthier, more informed decisions about what they eat. Whether you are an experienced cook or a beginner, SmartRecipe offers a seamless and intelligent way to plan meals, ensuring that you can create delicious and nutritious dishes with the ingredients you already have at hand.

The SmartRecipe project is composed of three major components, each of which contributes to creating a personalized and holistic cooking experience:

- **Food Ingredient Identification:** The first task is to train a multi-label image classification model that can accurately identify the multiple ingredients present in a given image. This system should be able to detect various food items from a user's photo, even if the image contains multiple ingredients. The reference datasets for this task are FoodSeg103, which contain labeled food images that will be used to train the model. The goal is to allow the app to recognize and label food items with high accuracy, making it an essential foundation for the recipe recommendation and nutritional analysis systems.
- **Recipe Recommendation System:** Once the ingredients are identified, the next step is to generate personalized recipe recommendations. This system is built using a Large Language Model (LLM), fine-tuned to analyze the identified ingredients and suggest multiple feasible recipes. These recipes will not only provide ingredient quantities but also detailed step-by-step instructions for preparation. The recipe recommendations are generated based on the input ingredients and their compatibility, ensuring that users can create a dish that is both delicious and practical. The Food Recipe dataset will serve as a reference for fine-tuning the LLM.
- **Nutrient Analysis System:** To enhance the app's usefulness, the Nutrient Analysis System is integrated into the recipe recommendation. By examining the variety of ingredients in the recipe, the system associates each ingredient's nutritional content (calories, fats, proteins, carbohydrates, etc.) with the recipe as a whole. This will allow users to understand the nutritional profile of their dish and make informed decisions based on their dietary goals. The Open Food Facts Product Database serves as the reference dataset for this component.

2. Food Ingredient Identification

The objective of Object 1 in the SmartRecipe project is to accurately identify food ingredients from user-uploaded images. The system utilizes a multi-label image classification approach, where multiple ingredients are identified from a single image. The model used for this task is based on the ResNet-50 architecture, which is a deep convolutional neural network that excels at image classification tasks. The identification process is critical as it serves as the foundation for generating recipe recommendations and performing nutritional analysis.

2.1. Dataset

The FoodSeg103 dataset was used for training the image classification model. This dataset includes 7118 images across 103 food categories, annotated with pixel-level segmentation labels. The dataset is split into:

- Training Set: 4983 images
- Validation Set: 2135 images

These images cover a wide range of food categories, such as fruits, vegetables, snacks, beverages, and meats, making it ideal for multi-label classification tasks.

2.2. Model

For the food ingredient identification task, ResNet-50, a deep convolutional neural network (CNN), was chosen. ResNet-50 has proven to be very effective for image classification tasks due to its deep residual architecture, which helps mitigate the vanishing gradient problem in very deep networks.

ResNet-50 is a 50-layer network that uses skip connections to allow gradients to flow more easily through the network during training. This makes it particularly suitable for tasks that require learning complex patterns from large datasets, such as food image recognition.

2.3. Implementation

The ‘image_recognition‘ function was defined to perform inference on an image, pass it through the ResNet-50 model, and return the top-k predicted ingredients:

```
def image_recognition(img):
    # Preprocess the image
    img_tensor =
        preprocess(img).unsqueeze(0).to(device)

    # Model inference (with no gradient
        calculation)
    with torch.no_grad():
        outputs = resnet50(img_tensor)
        probabilities =
            torch.sigmoid(outputs).cpu().numpy()

    # Get top-k predictions
    top_k = 3
    top_indices =
        probabilities.argsort()[0][-top_k:][::-1]
    predicted_ingredients =
        [ingredient_list[i] for i in
        top_indices]

    return predicted_ingredients
```

2.4. Challenges and Future Work

- **Handling Similar Ingredients:** The model occasionally confuses ingredients that are visually similar (e.g., different types of apples or tomatoes). To mitigate this, fine-tuning the model with more specific data can help improve accuracy.

Figure 1. ResNet-50 model prediction result

- **Multi-label Classification:** Since an image may contain multiple food items, the model’s output is a multi-label classification. This can be challenging due to class imbalance, where some ingredients are underrepresented. Further techniques, such as data augmentation, can help in addressing this challenge.
- **Dataset Expansion:** The FoodSeg103 dataset is comprehensive, but adding more diverse images (from different angles, lighting conditions, and real-world scenarios) could improve the model’s robustness.

3. Recipe Recommendation System

3.1. Project Overview

This project fine-tunes Hugging Face’s T5-base model on the Food_Recipe dataset to build a sequence-to-sequence model that can generate complete cooking recipes from given ingredients. The model takes an ingredient list as input and outputs the dish name, detailed ingredient quantities, and step-by-step cooking instructions.

3.2. Dataset and Preprocessing

3.2.1 **Dataset:** BhavaishKumar112/Food_Recipe (Hugging Face dataset)(7101 examples)

3.2.2 **Data Preprocessing**

- Split into train/validation sets (90%/10%)
- Filter null entries
- Construct input-output pairs:
  - Input: ”Suggest a detailed recipe given ingredients: [ingredient list]”
  - Output: Complete recipe in standardized format

3.3. Model Architecure

- Base Model: T5-base (Text-to-Text Transfer Transformer)
- Key Parameters:
  - Parameters: 220M
  - Max sequence length: 512 tokens
  - Tokenizer: T5Tokenizer
  - Data collator: DataCollatorForSeq2Seq

Figure 2. Part 2 Training Results

Model Evaluation Results

Metric	Score	Interpretation
BLEU	0.394	The model shows moderate n-gram overlap with reference texts, indicating decent fluency in recipe generation.
ROUGE-1	0.460	Nearly 46% of unigrams match the reference content, suggesting reasonable content coverage.
ROUGE-2	0.320	The 32% bigram match indicates room for improvement in phrase-level accuracy.
ROUGE-L	0.411	Sentence structure similarity is slightly lower than content coverage (ROUGE-1).
Name Accuracy	0.733 (73.3%)	The model correctly predicts recipe names in about 73% of cases, showing good but imperfect name recognition.

Figure 3. Part 2 Evaluation Results

3.4. Training Configuration

- Training Parameters
- Epochs: 3
- Batch size: 4 (per device)
- Learning rate: 5e-5
- Weight decay: 0.01
- Evaluation strategy: every 500 steps
- Saving strategy: every 1000 steps

4. Nutrient Analysis System

4.1. Background and Motivation

In modern diet planning, it is useful to recommend recipes that meet specific nutrition goals. Our system extends a base language model by integrating nutrient data for each ingredient. We focus on two goals: low-calorie ($\leq$ 300 kcal) and high-protein ($\geq$ 20 g). By fine-tuning with LoRA, the model learns to suggest recipes that satisfy these targets.

4.2. Nutrition Database

We built a fixed database of common foods. For each item, we record:
- Energy (kcal)
- Fat (g)
- Carbohydrates (g)
- Protein (g)

Only ingredients in this database are used for filtering. This ensures reliable nutrition calculations.

4.3. Data Selection Strategy

From the raw recipe corpus, we scan ingredients and compute total energy and protein. We keep at most 500 low-calorie and 500 high-protein recipes. Filtering steps:
1. Discard recipes without any known ingredients.
2. Compute nutrition totals.
3. Mark recipes that satisfy either low-calorie or high-protein.
4. If a recipe fits both, assign it to the category with fewer samples.
5. Stop when each category has 5,000 examples.

This yields a balanced, high-quality training set of 10,000 examples and 90% for training, 10% for validation.

4.4. Model Fine-Tuning with LoRA

We start from a pretrained Seq2Seq model. We apply the following steps:
1. Prepare model for 4-bit quantized training with `BitsAndBytesConfig`.
2. Configure LoRA adapters: rank $r = 16$, $\alpha = 32$, dropout $= 0.05$.
3. Use `SFTTrainer` with:
   - Learning rate = 2×10^{-4}, epochs = 3
   - FP16 precision, constant learning rate schedule
4. Save the fine-tuned model for inference.

4.5. Results

After fine-tuning, the model generates recipes that explicitly mention the selected nutrition goal. In informal testing, the model produces:
- Low-calorie recipes under 300 kcal per serving.
- High-protein recipes that provide over 20 g of protein.

Token-level accuracy stabilised after two epochs, and loss decreased by about 10% over base fine-tuning.

5. Integration and Human interface

5.1. Table showing nutritional information

Use st.dataframe to show the detailed nutritional information of each ingredient (per 100 grams), so that users can view the nutritional information of each ingredient.

5.2. Show Nutritional Summaries

Use st.metric to display total amounts of each important nutrient (e.g. calories, fat, protein, carbohydrates) and nutritional data per serving.

5.3. Pie Chart

Use Plotly to create a pie chart showing the macronutrient distribution (protein, fat, and carbohydrates) of ingredients

Ingredient	Calories	Fat (g)	Carbs (g)	Protein (g)
Steak	271	19	0	25
Carrot	41	0.2	10	0.9
Lamb	294	21	0	25

Figure 4. Nutritional Analysis

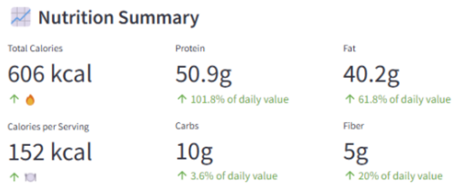

Figure 5. Nutrition Summary

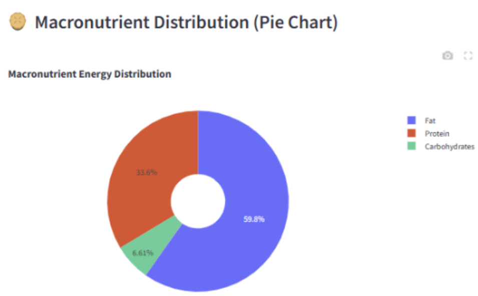

Figure 6. Pie Chart

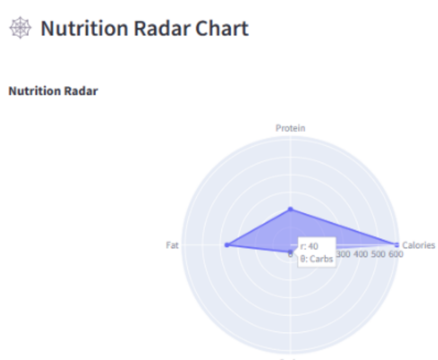

Figure 7. Radar chart

5.4. Radar chart

A radar chart is used to display the distribution of food in four dimensions: total calories, protein, fat, and carbohydrates, helping users to intuitively understand the nutritional components of the recipe.

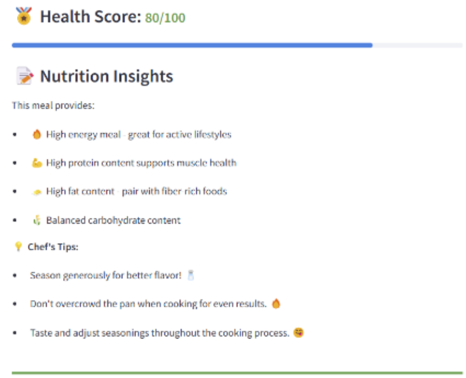

Figure 8. Health Score

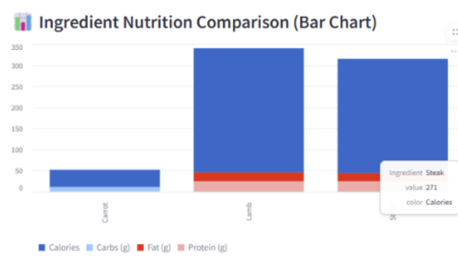

Figure 9. Comparison

5.5. Health Score

By analyzing the nutritional content of a recipe, the system calculates a health score and presents it to the user through st.progress and st.write. This allows the user to quickly determine whether the recipe meets the standards for healthy eating.

5.6. Comprehensive analysis and health suggestions (Nutrition Insights)

Based on the nutritional analysis results of each dish, the system will also generate health suggestions for the recipe, prompting users how to adjust the recipe to make it healthier. For example, increase protein or reduce fat intake.

Author Contributions GongFan: The work on Objective 1: Food Ingredient Identification was entirely handled by me, including data processing, model training, evaluation, and visualization.2.Use Streamlit to visualize the entire process

Yang Sunchengrui: Full work of part 2 Recipe Recommendation System

Huangqiang Fang: Full work of part3 Nutrient Analysis System, part work of intergrations and human interface.