Decomposing Food Images for Better Nutrition Analysis: A Nutritionist-Inspired Two-Step Multimodal LLM Approach

Supplementary Material

A. Datasets Detail

Nutrition5k is a dataset that includes visual and nutritional data collected from Google cafeterias using a custom scanning rig. In contrast, Gindee121 is a dataset sourced from our platform and annotated by nutritionists. The differences between these two datasets are demonstrated through sample images in Figure 10.

Table 4 presents the dataset information, while Figure 9 illustrates the distribution comparison between Nutrition5k and Nutrition320. Our proposed sampling method ensures that the distribution remains consistent across both datasets.

Dataset	Image Count	Average Value				
		Cal (kcal)	Weight (gram)	Protein (gram)	Carb (gram)	Fat (gram)
Gindee121	121	291.8	-	15.7	27.1	13.3
Nutrition320	320	251.8	239.7	18.0	20.4	11.7
Nutrition5k	3,241	249.2	212.2	17.6	19.0	12.4

Table 4. Dataset Information

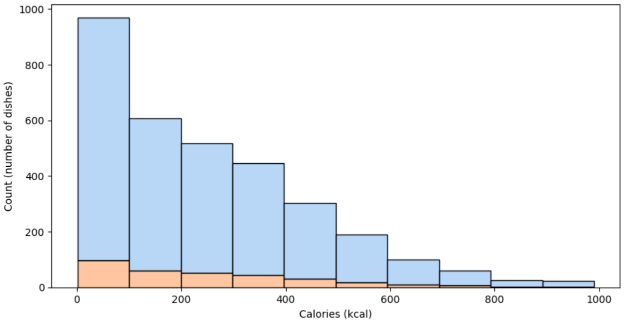
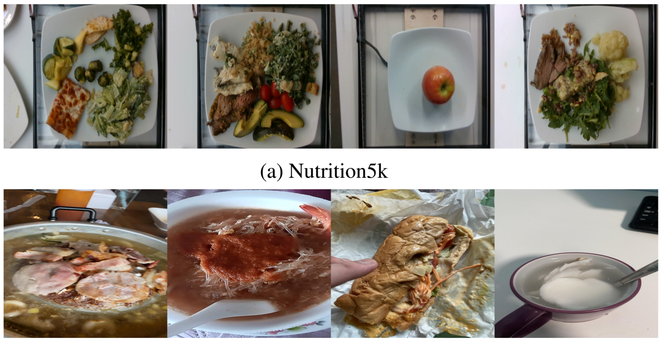


Figure 9. Calorie distribution of Nutrition5k (blue) with the selected 10% subset (Nutrition320), highlighted in orange.



(a) Nutrition5k

(b) Gindee121

Figure 10. Sample images from the Nutrition5k and Gindee121 datasets.

Prompt: The given image depicts food. Classify the image as a whole based on the following attributes, considering the most prominent features if the image contains multiple dishes. Output the results in JSON format. Use the provided restricted classes, and classify as "Other" if an attribute does not fall into the given categories.

- **Name:** Name of the most prominent or representative food item in the image.
- **Preparation Method:** Cooked, Raw, Processed, Baked, Fermented, Roasted, Other.
- **Cultural/Regional Origin:** Southeast Asian, East Asian, American, Thai, Filipino, Chinese, Global, Japanese, Korean, Indian, Western, Middle Eastern, Mexican, Italian, Greek, Other.
- **Food Type:** Main Course, Side Dish, Dessert, Drink, Soup, Ingredient, Other.
- **Food Category:** Meat, Vegetable, Grain, Seafood, Fruit, Dairy, Noodles, Other.
- **Function:** Breakfast, Lunch, Dinner, Snack, Supplement, Other.
- **Food Group (USDA):** Grains, Protein, Vegetables, Fruits, Dairy, Other.
- **Dietary Restriction:** Vegan, Vegetarian, Gluten-Free, Lactose-Free, None, Low Fat, Low Sodium, Other.
- **Food Allergens:** Soy, Dairy, Wheat, Shellfish, Eggs, Peanuts, Nuts, Fish, None, Other.
- **Packaging:** Restaurant, Homemade, Ready-to-Eat, Takeout, Frozen, Processed, Canned, Other.
- **Portion:** Single Dish, Snack, Meal, Other.
- **Camera View Angle:** Overhead, Close-Up, Side, Slight Angle, Other.

Output the attributes as a single JSON object summarizing the image as a whole.
Example JSON output format: {

```
  "Name":  "Pad Thai",
  "Preparation Method":  "Cooked",
  "Cultural/Regional Origin":  "Thai",
  "Food Type":  "Main Course",
  "Food Category":  "Seafood",
  "Function":  "Dinner",
  "Food Group (USDA)":  "Protein",
  "Dietary Restriction":  "Gluten-Free",
  "Food Allergens":  "Shellfish",
  "Packaging":  "Ready-to-Eat",
  "Portion":  "Single Dish",
  "Camera View Angle":  "Overhead",
}
```

Table 5. Gemini-1.5-Pro prompt for obtaining food details.

B. Prompts

B.1. Clustering Gindee

The prompts used to extract information for each food in the Gindee dataset to form Gindee121 are shown in Table 5.

B.2. Prompts Used in Each Experiment

We use standard prompting in Experiment 00 as the baseline, as shown in Table 6. Experiments 01 to 08, which apply visual prompting, have their prompt templates detailed in Table 10 (Prompt Structure) and Table 11 (Edited Prompt). Our proposed Two-Step Prompting is outlined in Table 7 (Step 1) and Table 8 (Step 2).

C. Nutrition Analysis Model Configs

From figure 12, these configurations are critical as directly affect the models' ability to process complex food descriptions and nutrition data.

Prompt: Please analyze the food or drink in the provided image and respond with a JSON object containing details of the most likely menu item. The JSON object should include the following structure:

```
{
  "ingredients_list_engfood":  [str],
  "weight":  int(percentage),
  "edible_percentage":  int(kcal),
  "oil_in_food":  float(gram),
  "cooking_method":  str,
}
```

The "ingredients_list_eng" field lists all major ingredients with their portion sizes in grams in English. If there are variations in minor ingredients like sauces or dressings, mention only one example, such as ”Korean sauce”, ”Sour Curry Paste”, or ”Spicy Salad Dressing”. Present this information as an array of objects, each containing the name and portion.
In the "weight" field, determine the weight of the standard serving size of the dish in grams. Ensure this is consistent with the listed ingredients.
For the "edible_percentage" field, calculate the percentage of the dish that is edible, excluding non-edible parts like packaging or inedible garnishes.
For the "oil_in_food" field, estimate the amount of oil absorbed in stir-fried and deep-fried dishes, in grams.
The "name_eng" field should contain the name of the food or drink. If the image contains multiple food items, include all the food names separated by commas in the "name_eng" field. If the exact name of the food is uncertain, provide a brief description of the dish based on its appearance and visible ingredients.
In the "cooking_method" field, identify the cooking method(s) used in preparing the dish, such as boil, poach, fry, grill, fresh, ready-to-eat, steam, or stir-fry.
The JSON object represents all the food in the whole image.

Table 6. Standard Prompting

Prompt: Please analyze the food or drink in the provided image and respond with a JSON object containing details of the most likely menu item. The JSON object should include the following structure:

```
{
  "food_description_eng":  str,
  "name_eng":  str,
  "weight":  int(gram),
  "cal":  int(kcal),
  "protein":  float(gram),
  "fat":  float(gram),
  "carb":  float(gram),
  "weight":  str,
  "health_suggestion_eng":  str,
  "ingredients_list_eng":  [str]
}
```

The ”food_description” field should provide a unique and detailed description of the food item, including its appearance, texture, and any notable characteristics.
The ”name_eng” field should contain the name of the food or drink. If the image contains multiple food items, include all the food names separated by commas in the ”name_eng” field. If the exact name of the food is uncertain, provide a brief description of the dish based on its appearance and visible ingredients.
When estimating the nutritional values, refer to reliable nutrition databases such as the USDA FoodData Central or other reputable sources. Use the data from these databases to calculate the ”weight”, ”cal”, ”protein”, ”fat”, and ”carb” based on the identified ingredients and their proportions in the dish.
In the ”weight” field, provide a brief recommendation on how the dish fits into a balanced diet or any potential dietary concerns based on the nutritional content.
If there are variations in minor ingredients (such as sauces or dressings), mention only one example, like ”Korean sauce”, ”Sour Curry Paste”, or ”Spicy Salad Dressing”.
The JSON object represents all the food in the whole image

Table 7. Two-Step Prompting (Step 1)

Prompt: Please analyze the food or drink in the provided image with the following food details to estimate its nutritional value and respond with a JSON object containing cal, protein, fat, and carb. Content: Food Detail:

{
food_detail
}

- Calculate the total ”cal” in kcal for the dish by aggregating the caloric content of individual ingredients listed in the ”ingredients_list_eng”. Use reputable nutritional databases such as USDA FoodData Central to establish accurate values based on quantity in grams.
- Estimate the total ”protein” content in grams, summing up contributions from ingredients that are significant protein sources.
- Determine the total ”fat” content in grams, taking into account all fatty components from the ingredients listed, including oils absorbed during cooking.
- Calculate ”carb” content in grams by summing the carbohydrate contribution of each ingredient, particularly from grains and vegetables.

Ensure your calculations are based on the aggregated input data provided, maintaining alignment with listed ingredients and their proportions. Present your findings in a JSON object representing the overall nutritional profile.

Table 8. Two-Step Prompting (Step 2)

Prompt:
Please analyze the food or drink in the provided image **PromptExp**$_i$. The ”food_description_eng” field should provide a unique and detailed description of the food item, including its appearance, texture, and any notable characteristics. The ”name_eng” field should contain the name of the food or drink. If the image contains multiple food items, include all the food names separated by commas in the ”name_eng” field. If the exact name of the food is uncertain, provide a brief description of the dish based on its appearance and visible ingredients. When estimating the nutritional values, refer to reliable nutrition databases such as the USDA FoodData Central or other reputable sources. Use the data from these databases to calculate the ”weight”, ”cal”, ”protein”, ”fat”, and ”carb” based on the identified ingredients and their proportions in the dish. In the ”weight” field, provide a brief recommendation on how the dish fits into a balanced diet or any potential dietary concerns based on the nutritional content. If there are variations in minor ingredients (such as sauces or dressings), mention only one example, like ”Korean sauce”, ”Sour Curry Paste”, or ”Spicy Salad Dressing”. The JSON object represents all the food in the whole image.

Table 10. Prompts template for i is experiments from **01** to **08** see the detail in Table 11.

PromptExp$_i$	**Prompt**
01, 02	“with food detection box”
03	“(original image on the left, food segmentation image on the right)”
04, 05	“(original image on the left, food segmentation inside bounding box image on the right)”
06	“(original image on the left, food segmentation image on the right)”
07	“(original image on the left, food semantic segmentation image on the right)”
08	“(original image on the left, food panoptic segmentation image on the right)”

Table 11. Prompt 01 to 08

Model	Temp	MaxOutToken	ContextLength
GPT-4o	0.5	1024	128K
Gemini-1.5-Pro	0.5	1024	1M
Gemini-2.0-Flash	0.5	1024	1M
Gemini-2.0-Pro	0.5	1024	2M
Qwen2.5-VL-72B	0.5	1024	32K
Llama3.2-90B-Vision	0.5	1024	33K
Gemini-2.0-Flash-Thinking	0.5	1024	1M
o1	0.5	1024	200K

Table 12. MLLMs inference configs for nutrition analysis

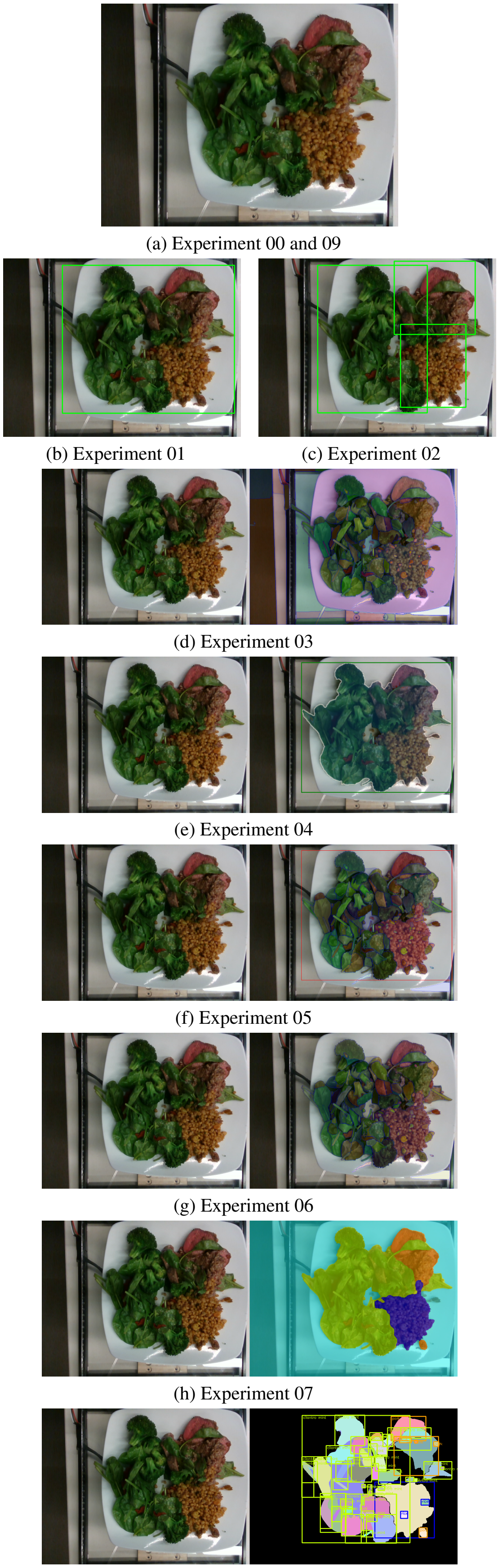
D. Visual Prompting Model Configs

For experiments 03, 05, and 06, we used “SAM2AutomaticMaskGenerator” with the configuration shown in Table 5. For experiment 04, we utilized “SAM2ImagePredictor” with the model *facebook/sam2.1-hiera-large* and bounding boxes from experiment 02. Both “SAM2AutomaticMaskGenerator” and “SAM2ImagePredictor” are sourced from SAM2-cookbook and for semantic and panoptic segmentation we use FoodSAM source from FoodSAM-cookbook.

Examples of the input image for each experiment are provided in Figure 11.

Parameter	Value
model	facebook/sam2.1-hiera-large
points_per_side	64
points_per_batch	128
pred_iou_thresh	0.7
stability_score_thresh	0.92
stability_score_offset	0.7
crop_n_layers	1
box_nms_thresh	0.7
crop_n_points_downscale_factor	2
min_mask_region_area	25.0
use_m2m	True

Table 13. SAM2.1 inference configs



(a) Experiment 00 and 09

(b) Experiment 01

(c) Experiment 02

(d) Experiment 03

(e) Experiment 04

(f) Experiment 05

(g) Experiment 06

(h) Experiment 07

(i) Experiment 08

Figure 11. Sample input image in each experiment.

E. Detailed Metrics

E.1. Mean Absolute Error

Mean Absolute Error (MAE) is a commonly used metric for measuring the average absolute difference between predicted and actual values. It is defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

where y_i represents the true nutritional value (e.g., calories, weight, protein, fat, carbohydrates) of a food item, $\hat{y}_i$ is the predicted value, and n is the total number of samples. A lower MAE indicates a more accurate prediction.

E.2. Jaccard Similarity

Jaccard Similarity is a statistical measure used to assess the similarity and diversity between sample sets. It is generally defined as the ratio of the intersection size to the union size, commonly referred to as Intersection over Union (IoU). Given two sets of ingredients, A and B, the equation for Jaccard Similarity is presented in Figure 10.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A|+|B|-|A \cap B|} \tag{2}$$

E.3. Average IoU

Average Intersection over Union (Average IoU) is a metric used in object detection to measure how well predicted bounding boxes align with ground truth boxes. It is computed as the mean IoU across all detected objects, where IoU is defined as the ratio of the area of overlap to the area of union between the predicted and ground truth boxes:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{3}$$

A higher Average IoU indicates better localization accuracy. see in Table 15.

F. Food Detection

To evaluate the model's understanding of food items and ingredients, we conduct a box detection experiment on the Nutrition320 dataset. Our approach follows the pipeline outlined in official cookbooks, such as gemini-cookbook and qwen-cookbook. The prompts used in this experiment are detailed in Table 14.

G. Ingredients Database

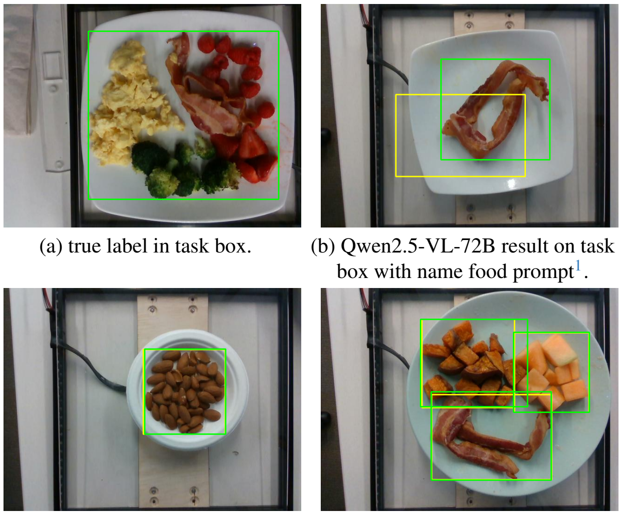
To evaluate the model's understanding of food ingredients, we employ the pipeline illustrated in Figure 7. For the nutrition database, we use the **United States Department of Agriculture (USDA)** database, which contains more than

Experiment	**Prompt**
food	“Detect the 2D bounding box of the food ”
food	“Detect the 2D bounding boxes of the food”
name food	“Detect the 2D bounding box of the food, **name**”
ingredients	“Detect the 2D bounding boxes of the ingredients”

Table 14. Detection Prompts, **name** is the variable of food name

Model	Experiment	Box Type	Average IoU
Gemini-2.0-Flash	food	Box	0.577
Gemini-2.0-Flash	name food	Box	0.590↑
Qwen2.5-VL-72B	food	Box	0.273
Qwen2.5-VL-72B	name food	Box	0.297↑
Gemini-2.0-Flash	food	Boxes	0.691
Gemini-2.0-Flash	ingredients	Boxes	0.690↓
Qwen2.5-VL-72B	food	Boxes	0.273
Qwen2.5-VL-72B	ingredients	Boxes	0.298↑

Table 15. Food Detection results on nutrition320



(a) true label in task box.

(b) Qwen2.5-VL-72B result on task box with name food prompt[1].

(c) Gemini-2.0-Flash on task box with food prompt[2].

(d) Gemini-2.0-Flash results on task boxes with ingredients prompt[3].

Figure 12. sample images of food detection on nutrition320.

1,500 ingredients. Furthermore, we consider **Nutrition5k** dataset as an alternative, which includes approximately 200 ingredients.