COSE474-2024F: Final Project Report
"Food Ingredient Detection and Recipe Recommendation Model"

Donghyun Kim

1. Instruction

As Artificial Intelligence (AI) continues to advance, it has become increasingly common to use AI like ChatGPT in everyday life. Many people rely on large language models (LLMs) like ChatGPT for tasks such as research and learning, while others use generative AI to create images or videos tailored to their preferences. However, in the area of "food consumption" which is closely tied to our daily lives, the impact of AI remains relatively limited. Tasks like creating entirely new recipes, which rely heavily on taste and sensory evaluation, appear to be beyond the current capabilities of AI. However, recommending existing recipes that align with an individual's available ingredients is a task well within the reach of current AI technology.

On a global scale, food waste poses a serious threat to food security and has become a critical issue to address. In fact, in 2022, households generated 631 million tons of food waste (United Nations Environment Programme, 2024). The developed model aims to naturally reduce food waste by offering personalized recipe recommendations tailored to the ingredients individuals already have. This approach may have a greater influence than merely emphasizing the need to reduce food waste, as it provides direct and practical solutions.

This project proposes the development of a model that automatically recognizes ingredients from images captured by a camera and recommends actionable food recipes based on the identified ingredients.

2. Methods

This project developed a model that provides recipes simply by taking a picture of available ingredients. The goal was to create a convenient system that eliminates the need for manual ingredient input or verifying if a recipe meets specific conditions. To implement this, the model integrates the object detection capabilities of YOLOv8 (Ultralytics, 2023) and the generative features of the LLaMA2 (Meta AI, 2023) Large Language Model (LLM). When an image of ingredients is provided, the fine-tuned YOLOv8 detects the food ingredients and extracts their labels. These labels are then reformatted into a predefined prompt and passed to the fine-tuned LLaMA2, which generates a recipe based on the detected ingredients.

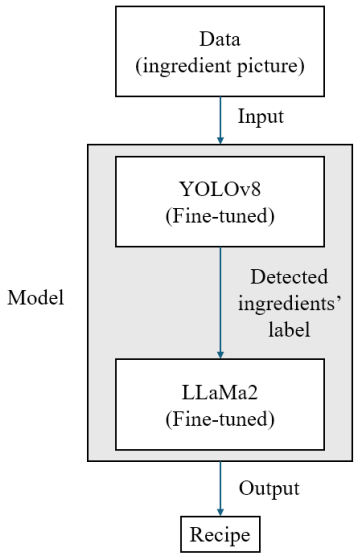


Figure 1. Model Figure

Algorithm 1 Model Algorithm

Input: Ingredient picture
Output: Generated recipe
LOAD YOLO_model and LLaMa_model
Target_image ← **READ** Ingredient picture
Results ← YOLO_model.predict(Target_image)
Label_list ← empty list
for all Result ∈ Results **do**
  APPEND Result to Label list
end for
Input_text ← "Ingredients: Label list"
Tokenized_input ← LLaMa_model.encode(Input_text)
Tokenized_output ← LLaMa_model(Input_text)
Output ← LLaMa_model.decode(Input_text)

3. Experiments

The model utilized two separate datasets, one for food ingredient images and another for recipes. First, the food ingredient image dataset was sourced from Roboflow's "food-recipe-ingredient-images-0gnku" dataset (2023). This dataset contains a total of 9,780 images and 120 food ingredient classes, including items such as Apple and Banana. The dataset was split into 86% for the training set, 8% for the validation set, and 6% for the test set.

For the recipe dataset, we used the "Shengtao/recipe" dataset (2022) available on Hugging Face. This dataset contains 32,722 recipe entries, including various details such as the dish name, estimated cooking time, and ingredients. From this dataset, we extracted the fields relevant to our model, namely 'title', 'ingredients', and 'directions' (cooking method). We processed 10,000 entries into the format: *Recipe:* {*'title'*} *Ingredients:* {*'ingredients'*} *Directions:* {*'directions'*} During training, we divided the dataset into 80% for the training set and 20% for the test set.

This experiment was conducted in the Google Colab Pro environment, which provides an A100 GPU with 40GB of RAM. For the YOLOv8 model, we utilized the "yolov8m" variant. The initial configuration included a batch size of 32 and a learning rate of 0.001, with a total of 70 epochs used to fine-tune the model on the food ingredient image dataset. This fine-tuning enabled the model to accurately identify food ingredients from input images.

For the LLaMA2 model, we employed the "Llama-2-7b-hf" variant. The model was fine-tuned on the recipe dataset using an initial batch size of 4, a learning rate of 5×10^{-5}, and 3 epochs. Additionally, the LoRA (Low-Rank Adaptation) technique was applied to train only low-dimensional additional parameters, making the process more efficient. Prompt engineering was employed to enhance the quality of the generated responses. The model's input prompt was structured as follows: *"You are an expert chef. Based on the given ingredients, suggest a recipe title and step-by-step cooking directions. Ingredients:* {*input_ingredients*} *Directions:"* This prompt was designed to guide the model in generating high-quality and contextually appropriate recipe suggestions.

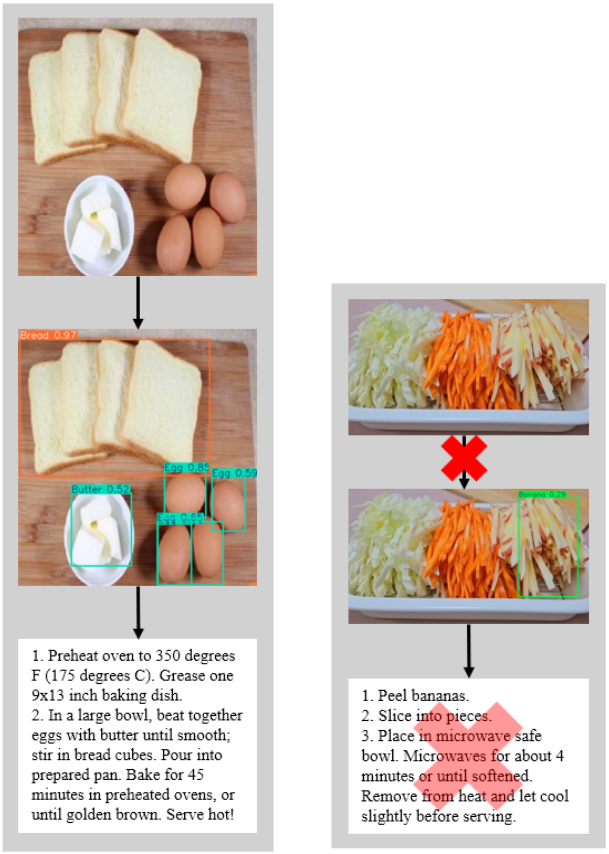
4. Results

The quantitative performance of the model after training is summarized in Table 1.

Additionally, to evaluate the practical performance of the model, it was tested on 40 samples. The criteria for distinguishing success and failure were divided into two phases. In the first phase, where the task was to identify ingredients from an image, the model's success was determined by whether it successfully detected all ingredients and correctly assigned the appropriate labels. In the second phase, which involved generating a recipe based on the recognized ingredients, success was defined by the model's ability to generate a recipe using the identified ingredients as the main components.

YOLOv8m (valid)	
Precision	0.68
Recall	0.534
mAP50	0.596
mAP50-95	0.372
LLaMa2	
Training Loss	0.7768
Validation Loss	0.807248

Table 1. Performance for YOLOv8m and LLaMa2

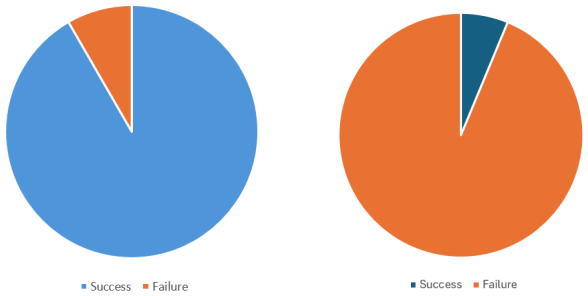


(a) Successful Case (b) Failure case

Figure 2. Qualitative Results

Out of 40 attempts, the model successfully generated recipes in 23 cases, as shown in Figure 2 (a), while failing to generate correct recipes in 17 cases. Failures in recipe generation,

such as the example in Figure 2 (b), were mainly due to misrecognition of ingredients, which led to incorrect recipe generation.



(a) OneType Case (b) ManyType Case

Figure 3. Object Detection Results

Furthermore, as shown in Figure 3, the model's ingredient recognition performance significantly declined when multiple types of ingredients were present in the image.

5. Conclusions

In conclusion, the method proposed in this report was not successful. The primary goal of this model—to provide recipes simply by taking a picture of the available ingredients—was heavily dependent on the ability to accurately recognize multiple ingredients present in a single image. However, the model failed to deliver satisfactory performance in this regard.

One potential reason for the poor performance in ingredient recognition lies in the selection of the dataset. Upon examining the training images in the dataset, it was observed that many images contained only a single type of ingredient. It was initially assumed that if the model could correctly recognize individual ingredients, it would also perform well when multiple ingredients were present in the same image. However, this assumption proved to be incorrect.

Additionally, another challenge was the significant variation in the appearance of the same ingredient. For example, the basic "apple" can appear in many forms—whole, julienned, or thinly sliced—and each variation differs in shape and color. The dataset failed to adequately account for such variations. This shortcoming became evident when julienned apples were misclassified as bananas, highlighting the limitations of the dataset's design.

6. Future directions

To improve the proposed method, the ingredient recognition process must be fundamentally revised. This requires modifications to the dataset. First, dataset should focus on Multi-Ingredient Images to better reflect real-world scenarios. The proportion of images containing multiple types of ingredients within a single image increased in the dataset. Second, Ingredient classes should be subdivided to account for variations in appearance. For example, the 'apple' class could be split into sub-classes such as 'whole apple,' 'julienned apple,' and 'sliced apple.' The model should be capable of recognizing the original ingredient regardless of its form—whether it is in its raw state, cut into rectangular shapes, or julienned.

However, there are practical challenges to achieving this. The world is filled with a vast variety of food ingredients, each of which can exist in nearly infinite variations. Developing a model that can recognize every possible variation of every ingredient is infeasible. Therefore, it is essential to establish clear criteria for determining which classes of ingredients and their variations should be included in the model.

References

Food waste index report 2024. United Nations Environment Programme, 2024.

AI, M. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://ai.meta.com/llama/.

Roboflow. Food recipe ingredient images, 2023. URL https://universe.roboflow.com/food-recipe-ingredient-images-0gnku/food-ingredients-dataset.

Shengtao. Recipe dataset, 2022. URL https://huggingface.co/datasets/Shengtao/recipe.

Ultralytics. Yolov8: State-of-the-art object detection model, 2023. URL https://github.com/ultralytics/yolov8.

(UNE, 2024) (Roboflow, 2023) (Shengtao, 2022) (Ultralytics, 2023) (AI, 2023)