Dean&Francis

ISSN 2959-6157

Automated Household Food Management and Recipe Recommendation System Based on Visual Recognition and LLM Knowledge Base

Jingwei Zhang

SWJTU-Leeds Joint school computer science, Leeds University, Leeds, UK

Email: sc21jz2@leeds.ac.uk

Abstract:

With continuous economic development and intensifying competition, individuals increasingly face the conflict between the pursuit of a better life and the demand for efficiency. This conflict is particularly pronounced in the realm of dietary habits. A healthy and well-balanced diet often requires significant time investment in planning and decision-making, yet people frequently lack sufficient time for such considerations. As a result, the need for an efficient system to assist with meal planning and food management has become more apparent. To address this challenge and help individuals achieve a balance between healthy eating and lifestyle efficiency, we have developed an easy-to-use, natural language-based automated household food management system leveraging visual recognition technology, large language models (LLM), and knowledge base technologies. The system automates household food management tasks, including inventory input, stock display, expiration monitoring, and food output. Additionally, it customizes personalized recipes based on factors such as the current time, number of family members, taste preferences, special dietary needs, and available ingredients. According to user surveys, over 70% of respondents recognized the necessity of the system, and its recipe design received an average rating of 3.75 out of 5, indicating that the majority of users found the system's recipe recommendations acceptable.

Keywords: Visual recognition, large language models (LLM), knowledge base, food management, automation

1. Introduction

The conflict between quality of life and efficiency has become increasingly apparent with societal progress and economic development, causing many people to struggle with this issue. Existing research on this topic primarily focuses on two areas: personalized menu recommendations and automatic inventory management. In the area of personalized menu recommendations, relevant work includes Chaudhary et al.'s "Generating Indian Recipes with AI Algorithm"[1], Faisal et al.'s "Diet-Right: A Smart Food Recommendation System"[2], Jill et al.'s "Intelligent Food Planning"[3], and Luca et al.'s "Automatic Reasoning Evaluation in Diet Management Based on an Italian Cookbook"[4], all of which employ AI technologies for recipe recommendations and have achieved notable results. In the field of automatic inventory management, studies such as Khan's "IoT Based Grocery Management System: Smart Refrigerator and Smart Cabinet"[5] and Fujiwara's "A Smart Fridge for Efficient Foodstuff Management with Weight Sensor and Voice Interface"[6] have made certain advances.

These works demonstrate a gap in easy-to-use, low-cost automated management systems for household food management, and the high cost, complexity, and inflexibility of personalized menu recommendation systems make them difficult to apply in everyday life. These systems struggle to adapt to variable conditions, such as fluctuating household sizes, diverse dietary preferences, or specific nutritional requirements (e.g., low-calorie, high-protein meals), and they often fail to provide detailed cooking instructions alongside the recipe suggestions.

To address these challenges, this study proposes an automated household food management and recipe recommendation system based on visual recognition, large language models (LLMs), and knowledge base management. The system aims to integrate these technologies to provide user-friendly food inventory management, including features such as stock tracking, expiry monitoring, and personalized recipe recommendations based on real-time inventory and household preferences.

The implementation utilizes visual recognition and LLMs to continuously update a knowledge base with information on food inventory and customized preferences. The knowledge base supports specialized inventory management and recipe recommendations, enabling the LLM to offer expert recipe customization and cooking guidance, while enhancing inventory management efficiency. Ultimately, this system aims to meet the demands for ease of use, expertise, and flexibility in food management and menu planning.

Inspiration for this study's menu design approach comes from Goel et al.'s "Ratatouille: A Tool for Novel Recipe Generation"[7], Vassányi et al.'s "A Novel Artificial Intelligence Method for Weekly Dietary Menu Planning"[8], Wang et al.'s "Market2Dish: Health-Aware Food Recommendation"[9], and Min et al.'s "Food Recommendation: Framework, Existing Solutions, and Challenges"[10]. The use of prompt engineering in LLMs draws on Marvin's "Prompt Engineering in Large Language Models"[11] and Louie's "Prompt Engineering with ChatGPT: A Guide for Academic Writers"[12]. Additionally, the use of a knowledge base for LLMs was inspired by Taberko's "NLP and LLM-Based Approach to Enterprise Knowledge Base Construction"[13] and Li's "FlexKBQA: A Flexible LLM-Powered Framework for Few-Shot Knowledge Base Question Answering"[14].

2. Knowledge Base Design and Basic Information Overview

To provide readers with a clear understanding of the structure of this study, this section will focus on introducing the core design: the knowledge base. Subsequent sections will elaborate on how this design serves as the foundation for supporting additional functionalities related to automated food management and personalized menu customization. This approach enables readers to gain a clear, foundational perspective on the system's design.

2.1 Overview of the Knowledge Base

Before proceeding, it is necessary to introduce the knowledge base technology used in LLMs. A knowledge base is a collection of domain-specific knowledge and data designed to enhance the accuracy and expertise of LLMs in particular contexts. An accessible analogy would be that a knowledge base functions as a "cheat sheet" for LLMs, allowing them to answer questions in an open-book manner. For instance, a medical knowledge base can provide detailed drug usage guidelines, ensuring users receive accurate and timely health advice.

As shown in Figure 1, domain-related knowledge is first cleaned and vectorized before being stored in a vector database via an embedding model. When a user submits a query, it is also vectorized, passed through the embedding model, and used to retrieve the top k results from the vector database. These retrieved results, or "cheat sheet," are then combined with the user's query in the prompt, which is subsequently fed into the LLM to generate the final response based on the knowledge retrieved.

ISSN 2959-6157

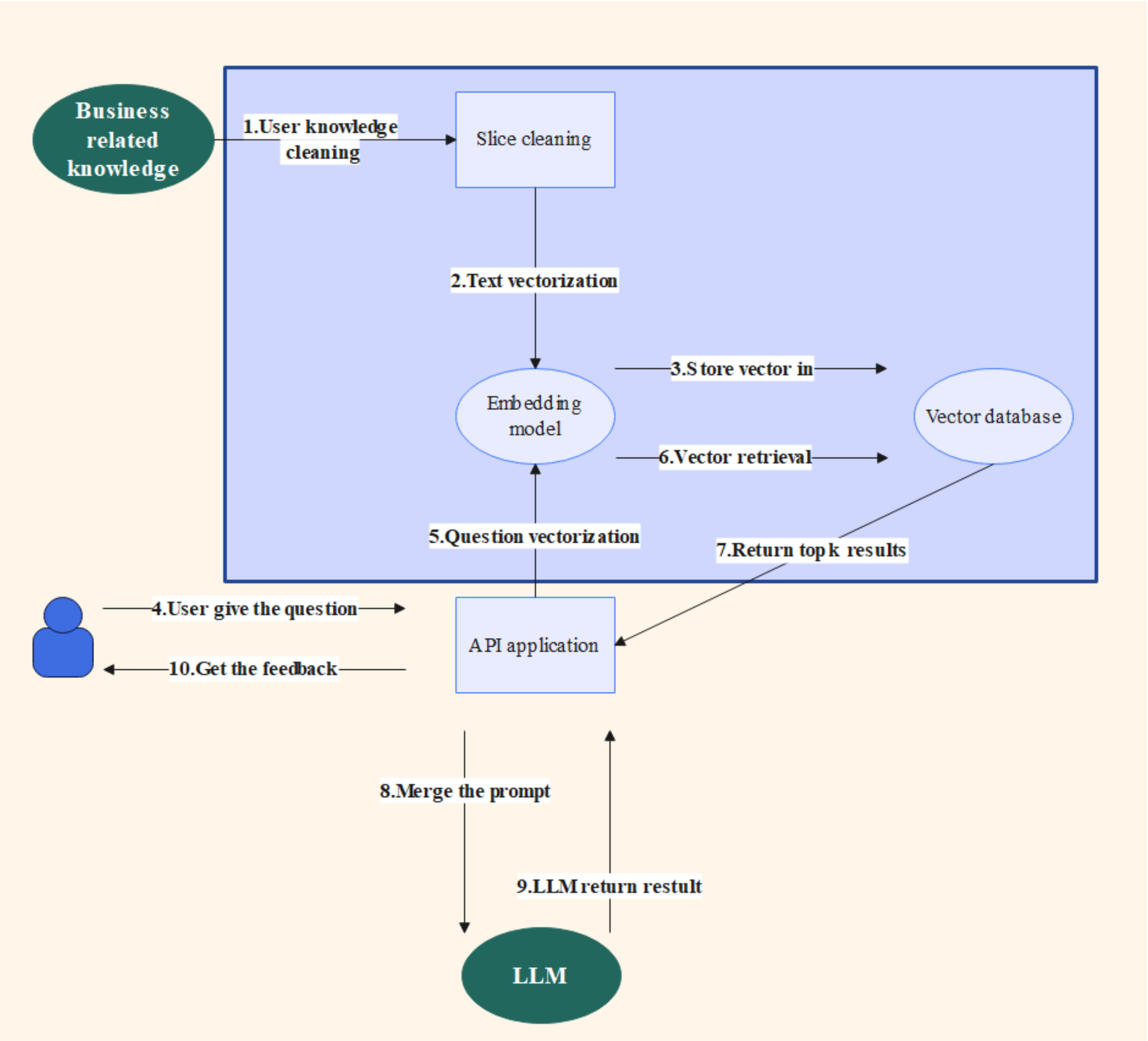


Figure 1 Knowledge base structure

2.2 Knowledge Base Design

In the design of this system, to ensure expertise in responses and enable text-based dynamic inventory management, knowledge base technology is employed. The structure of the knowledge base is shown in Figure 2 and consists of three main components. The first component is the inventory information, where the knowledge base serves as a database storing household inventory details, including item names, quantities, entry dates, and specific entry times. The second component stores the household's personalized menu preferences, such as the number of family members, preferred cuisine styles, and further customization needs. The third component contains specialized knowledge required for advanced recipe customization and inventory management. With the support of this knowledge base, the LLM can automatically generate a list of near-expiry items based on entry dates and expiration information, as well as professionally design menus and provide cooking instructions based on customization requirements and culinary knowledge.

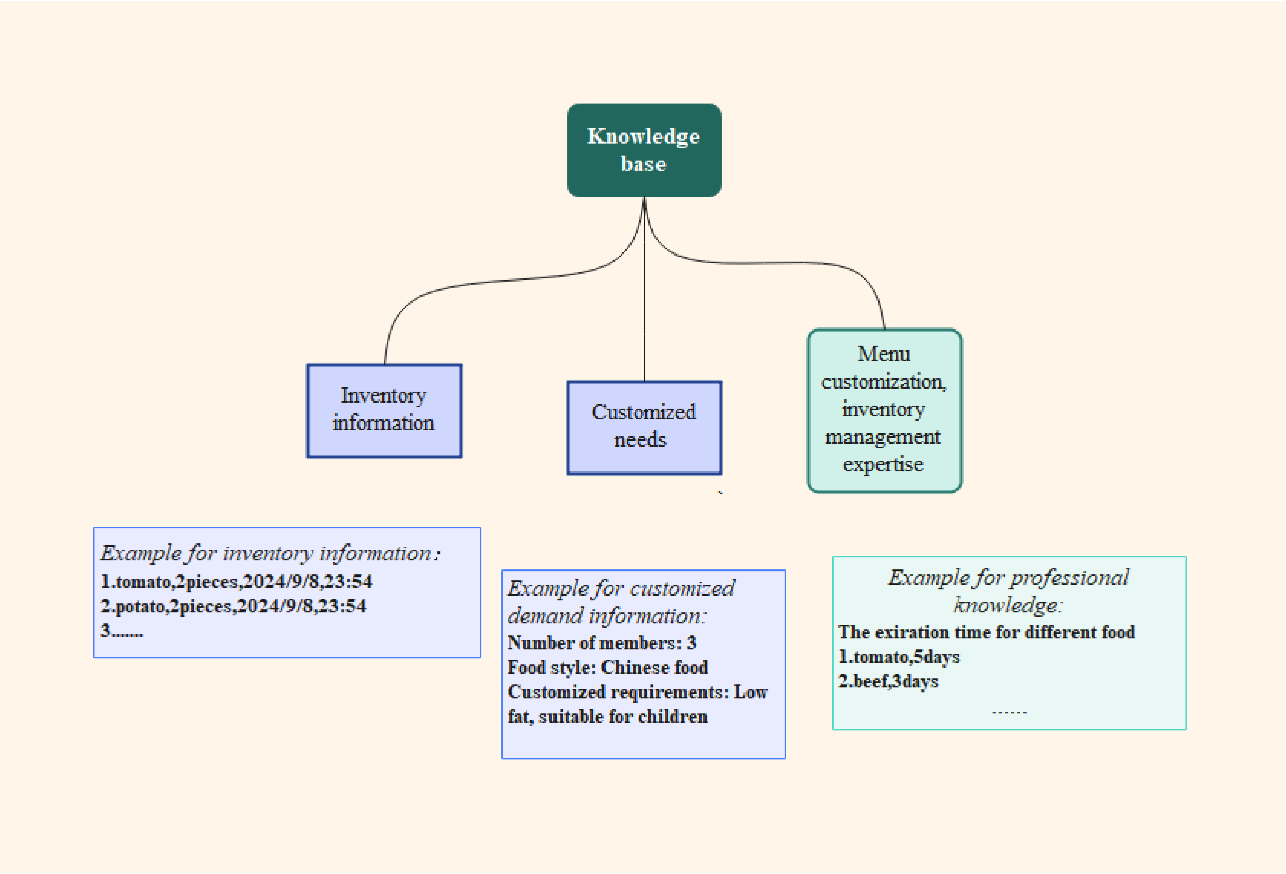


Figure 2 Knowledge files in system

3. Features and Implementation Details

Following the introduction of the knowledge base system, this section will address two main aspects: a descriptive overview of the system's features and a detailed explanation of the implementation. The implementation details include how to utilize visual recognition APIs for inventory management and how to update customization requirements and check inventory freshness.

3.1 Feature Overview

From the user's perspective, the system encompasses functionalities as depicted in Figure 3, categorized into two main areas: inventory management and menu customization. For inventory management, the system includes four core functions: Stock In, Stock Out, Freshness Monitoring, and Inventory Display. For menu customization, the system provides two key features: Generate Customized Recipes and Modify Customized Recipes.

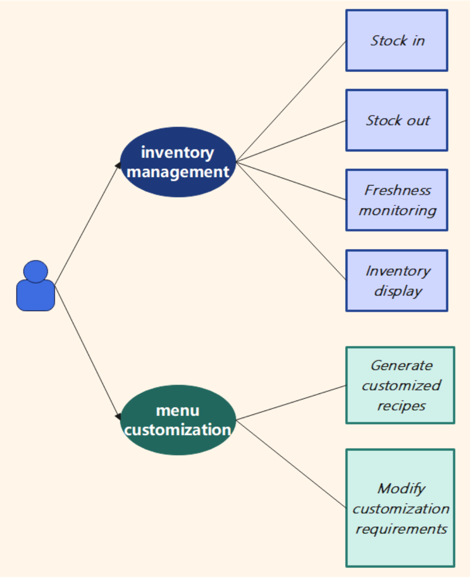


Figure 3 Functions of system from users' view

The functions will be further explained as follows:

The Stock In function utilizes visual recognition to identify items and automatically update the inventory records

ISSN 2959-6157

in the knowledge base. The process involves uploading a photo of the item to be identified. The system then determines the item's type and name. After reviewing the identification results, the user adds the item's weight or quantity, and the system automatically records the entry date and time, as illustrated in Figure 4.

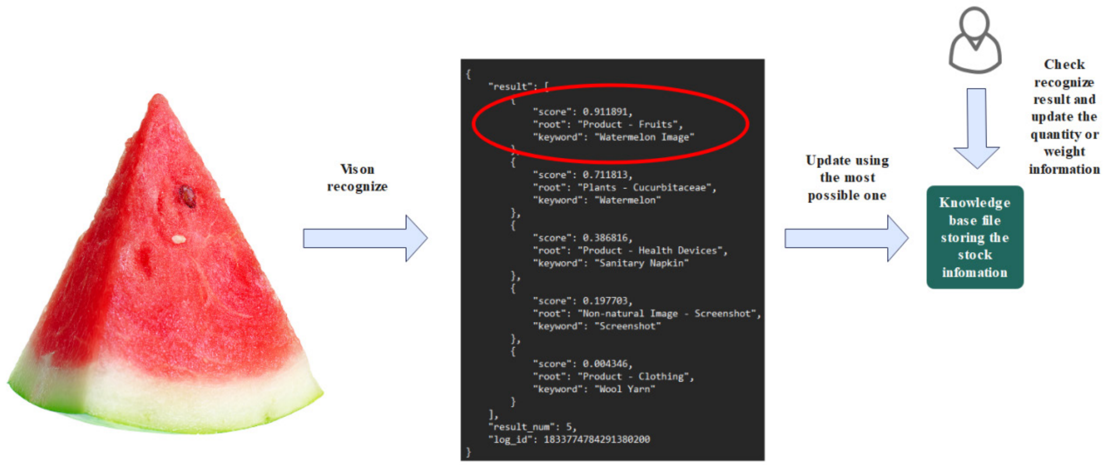


Figure 4 Stock in new item through vision recognition to knowledge base file

The Inventory Display function is a fundamental feature of the automated management system, providing information on currently stored items. As shown in Figure 5, it displays essential details such as item names, quantities, entry dates, and entry times. Notably, the system may list multiple records of the same item, such as two entries for "Beef." This design is intended to distinguish between items of the same type entered at different times, enabling precise management of identical items with different entry dates.

```
Items currently in the refrigerator:

1.Potatoes, 18 pieces, 2024/09/08, 23:25
2.Sausages, 4 pieces, 2024/09/08, 21:17
3.Tomatoes, 3 pieces, 2024/09/08, 17:54
4.Beef, 2kg, 2024/09/08, 17:54
5.Pork, 4kg, 2024/09/08, 17:54
6.Chicken, 1kg, 2024/09/08, 17:54
7.Cabbage, 2 heads, 2024/09/08, 17:54
8.Potatoes, 2 pieces, 2024/09/08, 23:26
9.Cabbage, 2 heads, 2024/09/08, 23:27
10.Beef, 2kg, 2024/09/08, 23:33
11.Beef, 2kg, 2024/09/08, 23:34
```

Figure 5 Result for listing all the inventory stuff command

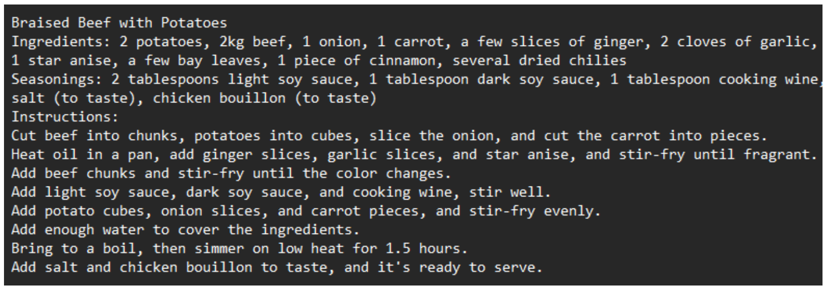
The Freshness Monitoring function, as the name suggests, automatically estimates the expiration dates of food items and alerts users to near-expiry or expired items. As illustrated in Figure 6, this functionality predicts food freshness and provides expiration alerts by examining the difference between the entry date and the current date

```
Food items at risk of expiration:

1.Potatoes, 2024/09/08, 23:25, about 180 days until expiration
2.Sausage, 2024/09/08, 21:17, about 14 days until expiration
3.Tomatoes, 2024/09/08, 17:54, about 7 days until expiration
4.Beef, 2024/09/03, 17:54, expired
5.Pork, 2024/09/03, 17:54, expired
6.Chicken, 2024/09/03, 17:54, expired
7.Cabbage, 2024/09/08, 17:54, about 7 to 8 days until expiration
8.Potatoes, 2024/09/08, 23:26, about 180 days until expiration
9.Cabbage, 2024/09/08, 23:27, about 7 to 8 days until expiration
10.Beef, 2024/09/08, 23:33, about 1 to 2 days until expiration
11.Beef, 2024/09/08, 23:34, about 1 to 2 days until expiration
```

Figure 6 Result for listing all food expiration information command

Generate Customized Recipes is one of the core functions of the system, allowing for the creation of menus and cooking instructions based on current time, number of family members, preferred cuisine style, user-specific requirements, and available ingredients. An example is illustrated in Figure 7, where a menu is designed for a family of three with a preference for Chinese cuisine and a special requirement for fat reduction. The resulting menu includes three dishes: Braised Beef with Potatoes, Chicken and Cabbage Soup, and Mushroom and Vegetable Porridge. To meet the fat reduction goal, the menu features lighter dishes like Chicken and Cabbage Soup and Mushroom and Vegetable Porridge. Each dish comes with specific ingredient requirements and cooking instructions. Notably, all ingredients used, except for spices and condiments, are currently available in the inventory.



Braised Beef with Potatoes
Ingredients: 2 potatoes, 2kg beef, 1 onion, 1 carrot, a few slices of ginger, 2 cloves of garlic, 1 star anise, a few bay leaves, 1 piece of cinnamon, several dried chilies
Seasonings: 2 tablespoons light soy sauce, 1 tablespoon dark soy sauce, 1 tablespoon cooking wine, salt (to taste), chicken bouillon (to taste)
Instructions:
Cut beef into chunks, potatoes into cubes, slice the onion, and cut the carrot into pieces.
Heat oil in a pan, add ginger slices, garlic slices, and star anise, and stir-fry until fragrant.
Add beef chunks and stir-fry until the color changes.
Add light soy sauce, dark soy sauce, and cooking wine, stir well.
Add potato cubes, onion slices, and carrot pieces, and stir-fry evenly.
Add enough water to cover the ingredients.
Bring to a boil, then simmer on low heat for 1.5 hours.
Add salt and chicken bouillon to taste, and it's ready to serve.

Figure 7 Result for giving menu in customized condition

Modify Customized Requirements ensures the system's flexibility and high level of customization. This function allows users to view and modify customization details, including the default number of family members, cuisine style preferences, and special requirements. By updating the relevant files in the knowledge base, the system can provide more adaptable menu customization.

Stock Out is addressed last because it involves two methods. The first method is manual, akin to removing items from a refrigerator. The second method is automatic and relies on menu design, which requires prior explanation of customized menu generation. Using the second method, the LLM can identify the quantity of ingredients needed for today's menu and automatically update the inventory. The process involves the LLM recognizing today's menu requirements, modifying the inventory knowledge base file accordingly, and uploading the updated file to the system for automatic inventory adjustment.

3.2 Implementation

During implementation, a range of tools from Baidu's Smart Cloud platform were utilized, including visual recognition APIs, LLMs, and associated knowledge base functionalities. For the LLM, we selected the ERNIE-4.0-8k model, which is the latest and highest-performing model available on the platform. This model has achieved impressive scores on various benchmarks, as detailed in Figure 8 and referenced in [15].

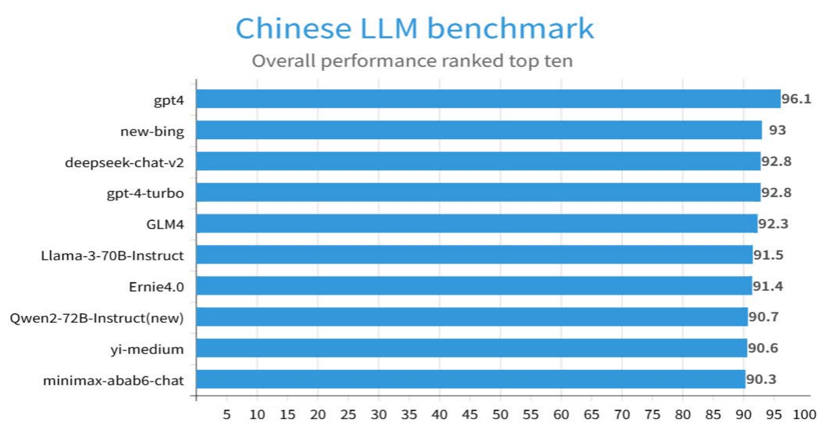


Figure 8 LLM benchmark based on Chinese corpora top 10

3.2.1 Stock In Using Visual Recognition, Stock Out, and Modify Custom Requirements

To enable efficient stock-in functionality, the system employs visual recognition technology, which has proven to be more efficient and consistent than manual data entry. For this purpose, we utilized the image recognition API from Baidu's Qianfan Smart Platform. After capturing and uploading images, the system performs batch recognition based on the uploaded photos and automatically suggests the most likely results. Users then review and verify the accuracy of the item type identification and manually input the quantity units. Upon completion, the system downloads the knowledge base file storing inventory information, adds the new data in the format of serial number, item name, quantity, entry date, and entry time, and re-uploads the updated knowledge base file. Figure 9 illustrates the process of inventory update.

ISSN 2959-6157

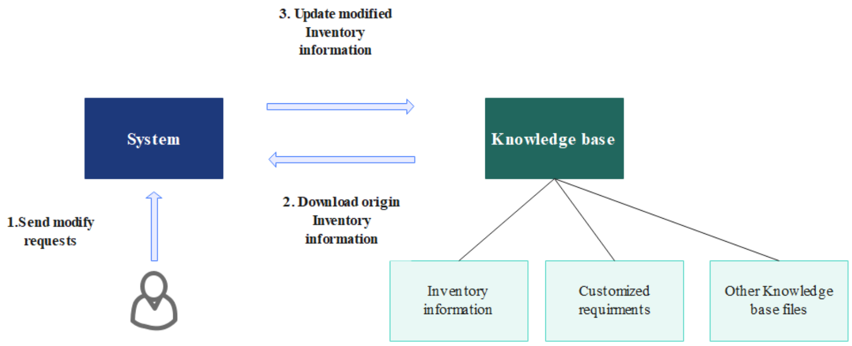


Figure 9 Knowledge base file update schematic

For the Stock Out function, the system similarly allows users to manually select items to be removed, after which it performs the same process of downloading, modifying, and re-uploading the knowledge base file. Additionally, an automated Stock Out feature based on the menu has been designed, requiring LLM assistance. The LLM's primary role is to identify ingredients used in the menu, download and adjust their quantities, and then update the inventory knowledge base file similarly to the manual stock out process. To enable the LLM to automatically recognize used ingredients and return the updated inventory file, a specialized prompt is employed. For example, the prompt might be: "You are an inventory manager, and your task is to identify the inventory materials used in the menu and update the inventory. For instance, if the menu uses two potatoes, reduce the inventory by two potatoes and return a new complete inventory list after deduction. Below is the menu: {menu_today}." This prompt ensures the LLM provides the desired output, and detailed prompt engineering will not be further elaborated here.

For the Modify Custom Requirements function, the same principle as Stock Out is applied. The system downloads the original customized requirements file, uses the LLM to make necessary updates, and then re-uploads the modified file to adjust details such as family size, cuisine style, and special requirements.

3.2.2 Customized Menu Design and Freshness Prediction

This section covers the implementation of two functionalities: menu customization and freshness management. Both functionalities rely on the dynamic updating capabilities of the knowledge base and its role in providing contextual information.

For menu customization, the system first checks the personalized requirements stored in the knowledge base and incorporates the current time into the prompts. This enables the LLM to understand these requirements and conditions, thereby generating a menu that meets the specified needs.

Similarly, for freshness monitoring, the LLM retrieves inventory information and freshness-related knowledge from the knowledge base prior to generating freshness reports. This information, as illustrated in Figure 10, includes item names and their preservation times. Based on this data, the LLM produces a freshness report with actionable insights.

Milk, expiration time: 7 days
Eggs, expiration time: 21 days
Bread, expiration time: 5 days
Yogurt, expiration time: 14 days
Apple, expiration time: 30 days
Banana, expiration time: 5 days
Chicken, expiration time: 2 days
Fish, expiration time: 3 days
Cheese, expiration time: 14 days
Tomato, expiration time: 7 days
Cucumber, expiration time: 5 days
Potatoes, expiration time: 60 days
Onions, expiration time: 30 days
Carrots, expiration time: 30 days
Ham, expiration time: 7 days
Sausage, expiration time: 14 days
Mayonnaise, expiration time: 90 days
Butter, expiration time: 60 days
Chocolate, expiration time: 180 days
Biscuits, expiration time: 120 days
Yogurt, expiration time: 14 days
Strawberries, expiration time: 3 days

Figure 10 examples for expiration knowledge stored in knowledge base

4. User Experience Survey Feedback and Scenario Analysis

To assess the system's menu design capabilities and market expectations, a survey and usage feedback collection were conducted. A total of 134 survey responses and 12 usage feedback submissions were obtained. The following summarizes key findings.

The survey was designed and distributed using Survey-

Monkey. It was divided into two parts:
The first part focused on the automated management functionality and included the following questions: a. Do you feel the need for an automated management system that handles food display, quantity, and entry date, even if it requires additional time for operation? Please rate your need on a scale from 1 to 5, where 5 indicates a strong need and 1 indicates no need. b. Do you require a system that alerts you about food freshness, even if it requires additional time for operation? Please rate your need on a scale from 1 to 5, where 5 indicates a strong need and 1 indicates no need. c. How do you perceive the future prevalence of a system that manages food and monitors freshness, especially if integrated with devices like refrigerators or smartphones? Rate the potential prevalence on a scale from 1 to 5, where 5 indicates very widespread and 1 indicates not widespread.
The second part evaluated the reasonableness and performance of menu design. Respondents were provided with customization conditions, including family size, time, cuisine style, current inventory, and special requirements, and were given a customized menu based on these conditions. Respondents rated the menu design quality on a scale from 1 to 5, where 1 indicates unacceptable and 5 indicates excellent. The menu design conditions are listed in Table 1, with a total of five cases.

Table 1 Diagram of recipe generation conditions in questionnaire survey

Number of family member	Current Time	Menu Style	Current inventory	Special requirements
1	7: 48	Chinese style	Chicken,Potatoes,Beef,Carrots, Pork,Tomatoes,Fish,Cabbage, Sausage,Onions	Reduce fat
2	12: 12	Western style	Lamb,Spinach,Bacon,Broccoli, Duck,Peppers,Turkey,Zucchini, Ham,Mushrooms	spicy
3	10: 12	Japanese style	Wagyu beef,Shiitake mushrooms,Chicken thigh ,Nori ,Miso paste,Daikon radish,Pork belly,Shiso leaves,Tofu,Edamame	Suit for child
5	19: 05	South Africa style	Lamb,Spinach,Bacon,Broccoli, Duck,Peppers,Turkey,Zucchini, Ham,Mushrooms	More meat
7	18: 23	Southeast Asia style	Chicken,Potatoes,Beef,Carrots, Pork,Tomatoes,Fish,Cabbage, Sausage,Onions	Sour flavor

Following the survey, feedback from actual system usage was also collected. The results from both the survey and usage feedback are summarized as follows:
For the first part:
Question a: 112 respondents rated the need for an automated management system above 3.
Question b: 98 respondents rated the need for a freshness monitoring system above 3.
Question c: 107 respondents rated the potential prevalence of such a system above 3.
The average scores were as follows:
Question a: 3.87,Question b: 3.78,Question c: 3.85
This indicates that over 70% of participants acknowledge the necessity of an automated management system.
In the second part, which evaluated menu generation:
The average scores for different conditions were 3.72, 3.63, 3.86, 3.81, and 3.74, respectively, with an overall average of 3.75. This reflects a high level of acceptance for the menu design functionality.
Regarding actual usage feedback, some users reported that the system effectively alleviated their difficulties in menu selection and the recommended dishes were satisfactory. However, others suggested improvements in the interface design and expressed hopes for future development.
Overall, the survey and feedback data indicate that the system is deemed necessary and demonstrates acceptable performance in menu design.

5. Conclusion

This study proposes a simplified automated home inventory management and recipe customization system based on LLM knowledge base technology and visual recognition. This system addresses a gap in home food inventory

ISSN 2959-6157

management and mitigates the conflict between efficiency and healthy, tasty eating. According to surveys and actual usage, over 70% of users acknowledge the system's necessity, and the menu design received an average rating of 3.75 out of 5, indicating that most users find the recipe design acceptable.

Despite the promising results, several market adaptation improvements are necessary for broader adoption. For instance, the cost of visual recognition is a significant concern. Frequent object recognition required for inventory management could lead to substantial costs if relying on external APIs. Ideally, local deployment of visual recognition could alleviate this issue. Additionally, there are challenges related to deployment and interface design. The current design is in its initial stages; therefore, a refined interface will be crucial for market adoption. Potential deployment platforms include mobile devices and refrigerators. Integrating computational capabilities into refrigerators could enable seamless execution of the system's functions and facilitate data collection for further analysis and application, aligning well with the system's objectives.

References

[1] Chaudhary, Smriti, et al. ChefAI.IN: Generating Indian Recipes with AI Algorithm. 13 Oct. 2022, https://doi.org/10.1109/tqcebt54229.2022.10041463.

[2] Faisal. "Diet-Right: A Smart Food Recommendation System." KSII Transactions on Internet and Information Systems, vol. 11, no. 6, 30 June 2017, https://doi.org/10.3837/tiis.2017.06.006.

[3] Freyne, Jill, and Shlomo Berkovsky. "Intelligent Food Planning." Proceedings of the 15th International Conference on Intelligent User Interfaces - IUI '10, 2010, https://doi.org/10.1145/1719970.1720021.

[4] Luca Anselma, et al. Automatic Reasoning Evaluation in Diet Management Based on an Italian Cookbook. 15 July 2018, https://doi.org/10.1145/3230519.3230595. Accessed 17 Aug. 2023.

[5] Khan, Muhammad Asad, et al. "IoT Based Grocery Management System: Smart Refrigerator and Smart Cabinet." 2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS), Dec. 2019, https://doi.org/10.1109/syscobiots48768.2019.9028031.

[6] Fujiwara, Masashi, et al. "A Smart Fridge for Efficient Foodstuff Management with Weight Sensor and Voice Interface." Proceedings of the 47th International Conference on Parallel Processing Companion, 13 Aug. 2018, https://doi.org/10.1145/3229710.3229727.

[7] Goel, Mansi, et al. "Ratatouille: A Tool for Novel Recipe Generation." IEEE Xplore, 1 May 2022, ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9814641.

[8] Vassányi, I., et al. "A Novel Artificial Intelligence Method for Weekly Dietary Menu Planning." Methods of Information in Medicine, vol. 44, no. 05, 2005, pp. 655–664, https://doi.org/10.1055/s-0038-1634022.

[9] Wang, Wenjie, et al. "Market2Dish: Health-Aware Food Recommendation." ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 17, no. 1, 16 Apr. 2021, pp. 1–19, https://doi.org/10.1145/3418211.

[10] Min, W., et al. "Food Recommendation: Framework, Existing Solutions, and Challenges." IEEE Transactions on Multimedia, vol. 22, no. 10, 2020, pp. 2659–2671, ieeexplore.ieee.org/document/8930090, https://doi.org/10.1109/TMM.2019.2958761.

[11] Marvin, Ggaliwango, et al. "Prompt Engineering in Large Language Models." Algorithms for Intelligent Systems, 1 Jan. 2024, pp. 387–402, https://doi.org/10.1007/978-981-99-7962-2_30.

[12] Giray, Louie. "Prompt Engineering with ChatGPT: A Guide for Academic Writers." Annals of Biomedical Engineering, vol. 51, 7 June 2023, pp. 2629–2633, https://doi.org/10.1007/s10439-023-03272-4.

[13] Taberko, V, et al. "NLP and LLM Based Approach to Enterprise Knowledge Base Construction." Bsuir.by, 2024, libeldoc.bsuir.by/handle/123456789/55618, https://libeldoc.bsuir.by/handle/123456789/55618. Accessed 11 Sept. 2024.

[14] Li, Zhenyu, et al. "FlexKBQA: A Flexible LLM-Powered Framework for Few-Shot Knowledge Base Question Answering." Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 17, 24 Mar. 2024, pp. 18608–18616, ojs.aaai.org/index.php/AAAI/article/view/29823, https://doi.org/10.1609/aaai.v38i17.29823.

[15] jeinlee1991. "GitHub - Jeinlee1991/Chinese-Llm-Benchmark." GitHub, 13 Apr. 2024, github.com/jeinlee1991/chinese-llm-benchmark. Accessed 11 Sept. 2024.