AI-Powered Smart Home Kitchen

Krisha Arora
Dept. of Information Technology
National Institute of Technology
Karnataka, Surathkal, Mangalore, India
krishaarora.231ai015@nitk.edu.in

Nidhi M
Department of Information Technology
National Institute of Technology
Karnataka, Surathkal, Mangalore, India
nidhim.231ai024@nitk.edu.in

Priyanka Nitin Mohorikar
Dept. of Information Technology
National Institute of Technology
Karnataka, Surathkal, Mangalore, India
priyankanitinmohorikar.231ai028@nitk.edu.in

Varshini Reddy Pateel
Dept. of Information Technology
National Institute of Technology
Karnataka, Surathkal, Mangalore, India
varshinireddy.231ai041@nitk.edu.in

Sona Mundody
Dept. of Information Technology
National Institute of Technology
Karnataka, Surathkal, Mangalore, India
sonamundody.217it005@nitk.edu.in

Ram Mohana Reddy Guddeti
Dept. of Information Technology
National Institute of Technology
Karnataka, Surathkal, Mangalore, India
profgrmreddy@nitk.edu.in

***Abstract*—Modern home kitchens face severe barriers in planning meals effectively, tracking ingredients, and managing inventory, resulting in food waste, overspending, and frustration for users. The paper presents a Smart Kitchen that applies artificial intelligence, computer vision, and natural language processing to revolutionize the way users interact with and manage their kitchen resources. A MobileNetV2-based object detection automatically detects and tracks ingredients, where items in the real world from fridge images are identified. The recipe suggestion framework is designed as a Constraint Satisfaction Problem to ensure compliance with ingredient availability and expiration constraints, giving priority to items which will spoil soon. A hybrid AI architecture is followed for intelligent meal recommendation, where GPT-2 is fine-tuned on culinary data combined with Retrieval Augmented Generation. This architecture will allow the generation of personalized contextually correct recipes, incorporating knowledge from external databases such as Spoonacular's recipe database. Furthermore, user input, such as dietary preferences for natural interaction, will be processed via a BERT-based semantic understanding layer. Thus, the UI enables hands-free operation with voice assistants while considering the experience of users in terms of diversity in accessibility. Generally speaking, Smart Kitchen reduces kitchen waste, makes it easier to decide what to cook, and provides a smoother cooking experience without making the process needlessly complicated.**

Keywords— Smart Kitchen, Ingredient Detection, MobileNetV2, Recipe Recommendation, Retrieval-Augmented Generation - RAG, GPT-2, Constraint Satisfaction Problem (CSP), Food Waste Reduction, Inventory Management, BERT.

I. Introduction

A lot of times, meal planning is time consuming. This results in repetitive meals, with much of the food ending up in the trash. The real problem is that ingredient availability and freshness are hardly ever checked in real time. Since there is no tool for efficient tracking of expiration dates, people often forget when their ingredients spoil. Human error makes traditional inventory management practices including hand-written lists prone to failure. This inefficiency forces people into buying items they already have or letting food rot, both of which have negative effects on the economy and the environment in turn.

According to FAO, approximately 1.3 billion tons of food produced is wasted annually, which is one-third of food produced around the world
citeb0. A large part of this food waste happens at the consumer level due to poor meal planning, overbuying, and ineffective tracking of ingredient freshness. Aside from the environmental impacts, this waste costs households economically through an increased cost of living. It also has adverse contributions ecologically through carbon emission, water usage, and landfills piling up. Hence, this paper seeks to minimize such food waste through the incorporation of smarter kitchen management systems, which will be crucial steps toward both economic savings and sustainability.

The challenge of solving this problem lies in the complexity of integrating multiple technologies for different parts of kitchen management. The big challenge is real-time recognition of different food items, which requires precise computer vision to work on casually taken images, several types of packaging, and a wide variety of food types present in the kitchen. It is similarly challenging to ensure seamless working without any issues among computer vision, NLP, speech recognition, and recipe databases. This is inclusive of the system understanding a user's context-such as matching ingredients already on hand with appropriate recipes, using items before expiration, and considering your dietary needs. And lastly, the interface should be set up in such a way that it works seamlessly for all, including users who have physical limitations.

Existing solutions address only particular parts of the problem, but they hardly offer a holistic and intelligent approach. For example, apps for inventory management based on barcodes, such as PantryCheck [2] and NoWaste [3], require users to scan items and update stock manually, increasing the chances of user error. Their systems also fail to automatically detect spoiled items in a traditional kitchen. Recipe recommendation sites or applications, such as AllRecipes [4] or Tasty [5], offer personalized recommendations for meals but do not integrate current ingredient availability and expiration

tracking. For example in the case of smart fridges, such as those developed by Samsung [6], these have internal cameras for tracking ingredients but are expensive.

This work develops a Smart Kitchen by integrating cutting-edge technologies: computer vision, AI, and NLP. The core concept is to create a single platform for real time ingredient recognition based on object detection using MobileNetV2, tracking freshness by formulating a CSP framework that suggests personalized recipes based on the availability of ingredients, dietary preferences, and expiration constraints. Moreover, meal recommendations are facilitated by a hybrid AI model using an integration of GPT-2 [7] and RAG [8] from the Spoonacular database. All of these are further enabled through voice and visual interfaces to make this system easily accessible for people with low technical experience or physical challenges. It points out that this work will minimize household food wastes by making smarter meal planning, using available ingredients.The system saves money while encouraging sustainable consumption habits by making the cooking process smooth, using ingredients before they spoil by suggesting personalized meals; it makes meal planning fun and easy. Therefore, this system allows easy access to and ease of use with multimodal interactions: voice, image, and text. Therefore, it is generally usable for diverse users ranging from students to busy professionals and senior citizens. Unlike traditional solutions, such as barcode based inventory apps [9] or smart fridges with few AI enabled features, the proposed system will be integrated with real time image-based ingredient detection, expiration monitoring, and personalized recipe suggestions. The proposed method is scalable and affordable. This offers hands free interaction along with an intelligent platform which truly transforms the way households manage their kitchens efficiently and sustainably.

II. Literature Survey

Various research efforts over the last few years have tried to provide solutions for different aspects of smart kitchen management, such as detecting food items and tracking inventory, all the way to AI-based recipe generation. However, these are usually individual functionalities, separate from a unified, interactive system that would cover all fields of kitchen automation.

The authors, Stefanus Christian et al., in [10], proposed a mobile application for food item and ingredient detection using deep learning from food images. Focuses on improving ingredient recognition accuracy, with no integration of either inventory management or recipe suggestion features within the system.

Summary: Marinis et al. [11] introduced a systematic review on machine learning techniques, mainly CNN-based, for the recognition of visual ingredients. It highlights challenges such as occlusion and visual complexity, but does not present an implemented system.

Authors in [12] proposed an Android mobile app that aims at reducing food waste by leveraging Rapid Application Development (RAD) methodology, Firebase Authentication, real-time database services, and the Spoonacular API. Its core functionality manages the ingredients inventories combined with personalized recipe recommendations considering the ingredients stored in one's kitchen.

While each of these contributions advances the field, none of them provide an end-to-end system that incorporates all of the elements: computer vision, AI-driven natural language processing, constraint satisfaction for expiration-aware meal planning, and multimodal user interaction. In contrast, our proposed Smart Kitchen system addresses this gap with a tightly integrated architecture comprising MobileNetV2 for food detection, a GPT-2 + RAG-based pipeline for generating personalized recipes, and CSP to ensure optimal ingredient usage before expiration.

TABLE I: Comparison of Related Works and Our System

Author(s)	Methodology	Strengths	Limitations
Stefanus Christian et al. (2022)	Food & ingredient detection using Deep Learning	Accurate ingredient recognition via computer vision	No inventory tracking, recipe generation, voice interaction, or automation
Marini's et al. (2023)	Visual recognition of food ingredients: a systematic review	Comprehensive review of ML based vision methods for ingredient detection	Not an implemented system; no real time features, recipe generation, or user interaction
Mei et al. (2024)	Android mobile application for recipe recommendation	Uses RAD, Firebase authentication, Database services and Spoonacular API	Inventory items are manually added, limited customization, no advanced AI
Proposed System (2025)	Integrated Smart Kitchen Assistant	Combines MobileNetV2, GPT-2 + RAG, CSP, and voice control	Currently relies on static fridge images, not live

III. Problem Statement and Objectives

Problem Statement:

Design an AI based smart kitchen that works in inventory tracking automatically, designs recipes based on expiration dates.

Objectives:

- Detect ingredients using OpenCV.
- Suggest recipes by using constraint satisfaction of expiration dates and GPT 2 generation.
- Allow hands free interaction by voice commands and image uploads.
- Recommend cooking tutorials.
- Reduce food wastage and improve kitchen related prep.

IV. Methodology

A. Overall System Architecture

The Smart Kitchen is a modular, AI powered pipeline that combines computer vision, constraint satisfaction and NLP for a fully automated cooking assistant. The goal of the system is

to assist users in making the best possible use of their available ingredients, reducing food waste, while also improving their cooking experience with creative recipe recommendations.

At a high level, the workflow of the system starts by acquiring the image, wherein a user snaps a snapshot of available food items kept inside the fridge. This image is then processed by a custom vision module based on the MobileNetV2 object detection algorithm, responsible for identification and classification of the food items present in the image.

From detection output, the system builds two key datasets:
Ingredient Set

$\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$: which contains all recognized ingredients.

Expiration Set

$\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$: where e_k denotes the predicted or known expiration date for ingredient i_k. These datasets are used with the Constraint Satisfaction model, which is used to filter recipes from a recipe database.

The filtered subset of recipes is then refined through a hybrid recipe generation module that combines fine tuned generative modeling using GPT 2 with Retrieval Augmented Generation (RAG) techniques. The RAG component connects with an external culinary API called Spoonacular to gather similar recipies for new recipe suggestions.

The final result a customized set of recipe recommendations is provided to user through a responsive front end interface that supports multimodal interaction via text, voice commands, and image input. Each recipe is displayed with step by step tutorial and is based on dietary preferences, cooking time or cuisine .

B. Constraint Satisfaction Problem (CSP) Formulation

The logic for recipe selection is modeled as a Constraint Satisfaction Problem. This mathematical representation makes sure that the selected recipes are not only feasible based on currently available ingredients but also optimized for prioritizing the usage of ingredients closest to expiration.

Let:

- $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$: the set of available ingredients
- $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$: the corresponding expiration dates
- $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$: the set of candidate recipes

Each recipe $r_j \in \mathcal{R}$ is defined by a subset of ingredients $\{i_k\} \subseteq \mathcal{I}$. Each ingredient variable i_k is binary: it is either available or unavailable.

A recipe r_j is deemed feasible if it satisfies the following two constraints:

1) Ingredient Availability Constraint

$$\forall i_k \in r_j, \quad i_k \in \mathcal{I}$$

All required ingredients must be available.

2) Expiration Validity Constraint

$$\forall i_k \in r_j, \quad e_k > \text{current\_date}$$

All required ingredients must not be expired.

Additionally, to reduce food waste, the system incorporates an Expiration Minimization Objective, which aims to prioritize recipes that use ingredients nearing their expiration. The urgency utility function is defined as:

$$\text{Urgency}(r_j) = \sum_{i_k \in r_j} \frac{1}{e_k - \text{current\_date} + 1}$$

This function assigns higher utility to recipes that consume ingredients with shorter remaining shelf lives. The optimization goal is to find the recipe r_j^* that maximizes the utility while satisfying the constraints:

$$r_j^* = \arg \max_{r_j \in \mathcal{R},\ \text{subject to constraints}} U(r_j)$$

This CSP based filtering mechanism ensures that only recipes which are both doable and urgency aware are passed on to the generation module.

C. Ingredient Detection using OpenCV and MobileNetV2

The system performs well depending on the correct ingredient recognition. The system will use an architecture from MobileNetV2 [13] for object detection in real-time. MobileNetV2 was further chosen due to its accuracy, speed, and flexibility of object detection performance in complex kitchen scenes with varied lighting and conditions.

OpenCV preprocesses the captured images, including resizing, normalizing the histogram for better quality. The results are input into the MobileNetV2 model which then outputs bounding boxes, class labels and confidence scores of each detected object.

A Non Maximum Suppression (NMS) step is applied to eliminate redundant overlapping detections. Then, the detected ingredient labels are standardized using a nutritional terminology, which maps various labels meaning essentially the same thing (e.g., "bell pepper" vs. "capsicum") to one unified ingredient taxonomy. In addition, expiration is inferred based on detection timestamp and typical storage conditions.

MobileNetV2, introduced by Sandler et al. [13], is a highly efficient convolutional neural network (CNN) [14] architecture specifically engineered for resource constrained environments such as mobile devices. It builds upon its predecessor, MobileNetV1, by introducing novel building blocks that significantly reduce computational complexity and memory while maintaining competitive accuracy for various computer vision tasks. The core innovations lie in its inverted residual blocks and linear bottlenecks.

Key Architectural Components:

The MobileNetV2 architecture is primarily composed of an initial full convolution layer then a series of stacked inverted residual blocks are used and a final classification layer. The building block called the inverted residual block (also known as a MobileNetV2 block), leverages the following principles:

1. Depthwise Separable Convolutions [14]: MobileNetV2 continues to utilize depthwise separable convolutions as an important element for computational efficiency. A standard convolution network performs filtering and combination of features in a single step. Depthwise separable convolution decomposes this into two distinct steps:

- Depthwise Convolution: A single filter is applied to each input channel independently. This step is responsible for spatial filtering.
- Pointwise Convolution [15] (1x1 Convolution): A 1x1 convolution is then applied to combine the outputs of the depthwise convolution across channels. This step handles channel wise feature combination.

This factorization significantly reduces the number of parameters and computational operations compared to standard convolution which makes the network smaller and faster.

2. Inverted Residual Blocks: Traditional residual blocks typically employ a "bottleneck" structure where the input feature maps are first compressed to a lower dimension which is then processed in that low dimensional space and finally expanded back to the higher original dimension for a skip connection that directly connects the higher dimensional input to the higher dimensional output. MobileNetV2 flips this paradigm and hence "inverted":

- Expansion Phase: It starts with a low dimensional input. A 1x1 convolution is used to expand this input to a much higher dimensional space. This expansion allows the network to operate on many feature representations.
- Depthwise Convolution: A depthwise convolution is then applied in this high dimensional space. This operation performs primary spatial filtering and feature extraction.
- Projection Phase (Bottleneck): Finally, another 1x1 convolution projects the features back down to a low dimensional output.
- Skip Connection: The residual connection is established directly between the narrow input and the narrow output of the block. This allows gradients to flow more easily through the network, mitigating the vanishing gradient problem in deep architectures.

Rationale behind this inversion is that "bottleneck" layers (low dimensional input/output) act as pivotal interface for information while high dimensional intermediate layer provides necessary capacity to learn complex non linear functions.

3. Linear Bottlenecks: One of the main design elements of MobileNetV2 is to use linear activation functions in the last projection layer, which is the 1x1 convolution that compresses down the channels. Most convolutional layers are typically followed by ReLU. They found this is suboptimal when these feature maps are projected to very low dimensional spaces. Applying a nonlinear activation such as ReLU to such "bottleneck" layers may lead to some loss in information. This is because ReLU cancels negative values thus discarding potentially valuable information in the compressed representation. A linear activation allows MobileNetV2 to preserve more information in general but most importantly in the low dimensional area, improving its representational power and thus overall accuracy. Non linearities are used, though specifically ReLU6, which is a ReLU with a clipped output after the expansion convolution and the depthwise convolution in the larger, intermediate layers.

Overall Structure

The MobileNetV2 architecture can be visualized as a sequence of these inverted residual bottleneck layers. The network typically begins with a standard convolutional layer, followed by multiple inverted residual blocks with varying expansion factors and output channels. A global average pooling layer and a final classification layer typically make up the network. MobileNetV2's new design, centered on inverted residual blocks and linear bottlenecks, effectively decouples the network's capacity (handled by the high dimensional expansion layer) from its input/output interface (handled by the narrow linear bottlenecks). This results in a highly efficient and accurate architecture that is well suited for deployment in mobile and embedded computing platforms where computational resources and power consumption are critical constraints.

D. Recipe Generation Enhancement

To provide users with unique, diffrent and nutritionally sound recipes, the system employs a hybrid recipe generation pipeline. The foundation represents a fine tuned GPT 2 language model on a domain specific corpus of more than 100,000 structured recipes which includes things like cooking techniques, ingredient combinations, cuisine types and dietary labels such as gluten free or vegan.

While GPT 2 is capable of offering recipes, it does not inherently check whether the information is factually correct or relevant. To fix this gap, within the system, a RAG framework is integrated. Upon receiving an ingredient list or user query (eg., "vegan lunch using spinach and tofu"), a BERT based [16] semantic embedding is generated and it captures the contextual intent of the query and retrieves top k most relevant documents from external sources (eg., the Spoonacular API [17]).

The documents serve as factual helpers and are combined with generative output through context aware transformers, effectively combining creativity with knowledge. The system ensures that only those recipes which align with CSP filtered ingredients and defined expiry constraints for final output.

The generated recipe includes:

• Title and a description
• Ingredient list
• Step by step instructions
• Video tutorial links if available

Recipes are presented to user through an interactive UI that supports feedback and it learns over time which creates a personalized and efficient cooking experience.

V. TRAINING THE MODEL

A. Training Environment

Training is conducted on Kaggle Notebook using the fruit-and-vegetable-image-recognition dataset from Kaggle. The environment took advantage of Kaggle's free GPU runtime with an NVIDIA Tesla T4, supported by 16 GB of RAM and 20 GB of short term storage. The software stack includes Python 3.10.1, TensorFlow 2.14 + and Keras as well as commonly used libraries for data processing and visualization such as NumPy, Pandas and Matplotlib.

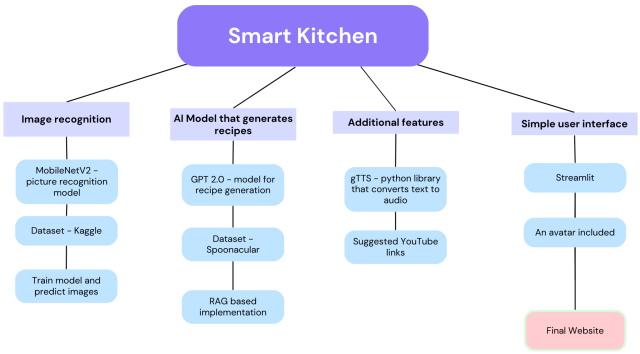

Fig. 1: Methodology block diagram

B. Training Image Recognition Model

A pretrained MobileNetV2 model (originally trained on the ImageNet dataset) is employed as the feature extractor to use transfer learning. There are 36 categories of fruits and vegetables in the dataset like apple, banana, carrot etc. Out of the total of 3,826 images out of which 3116 images (82%) for training, 358 images (9%) for testing and 352 images (9%) for validation, ensuring balanced evaluation across all classes.

The dataset is organized in a folder based split with each class stored in a separate subfolder under the folders for training, validation and testing. Images are first preprocessed and loaded with ImageDataGenerator, which resizes to 224×224 pixels, normalizes pixel values, and labels them automatically, depending on the folder names. The original classification layer of MobileNetV2 was removed and replaced with GlobalAveragePooling2D, a Dropout layer to reduce overfitting and a final Dense layer with softmax activation for classification across 36 categories. The base MobileNetV2 layers keeps its pretraining on ImageNet weights. It was compiled with the Adam optimizer, categorical cross entropy loss, and accuracy as the metric and fitted by the fit function, which iteratively updates weights over several epochs to enhance performance

Classification Model: MobileNet2

The study uses the MobileNetV2 model architecture, pretrained on the ImageNet dataset, with the top classification layer removed. On top of the frozen base, custom layers were added: a GlobalAveragePooling2D layer, followed by a Dense(128, ReLU) layer, and a final Dense(36, Softmax) layer for multi class classification. The input image size was fixed at 224×224×3, with training conducted using categorical cross-entropy loss, the Adam optimiser, and accuracy as the evaluation metric. Training was performed with a batch size of 32 for 20 epochs, while applying extensive data augmentation (rotation, shifting, shearing, zooming, and horizontal flipping) to the training set to improve generalisation.

C. Training RAG

User Input: The system takes a list of ingredients as input from the detected items from image recognition(e.g., ”chicken, garlic, lemon”).

Recipe Retrieval: It uses the Spoonacular API to fetch a list of recipes that include the provided ingredients. This provides relevant documents (recipes) based on the user’s input.

Embedding and FAISS Indexing: Each recipe (title, ingredients, and instructions) is processed to generate embeddings using a BERT model. These embeddings capture the semantic meaning of each recipe. A FAISS index is created to store these embeddings, enabling fast retrieval of similar recipes.

Query Handling: After retrieving recipes, the user can ask a query (e.g., ”How to cook chicken with garlic and lemon?”). The system converts the query into an embedding and performs a nearest neighbor search using FAISS to find the most relevant recipes based on the user’s query.

Recipe Generation: The retrieved documents (relevant recipes) are then fed into a GPT-2 model. The GPT-2 model generates a new recipe by incorporating both the provided ingredients and the instructions from the retrieved documents, producing a unique recipe with step-by-step instructions.

How RAG is Used:

Retrieval: The RAG framework starts by retrieving relevant documents (recipes) from the Spoonacular API and from the FAISS index. The Spoonacular Recipes are used to populate our vector database.

Generation: The GPT-2 model generates a new recipe based on the retrieved content (ingredients and instructions).

Augmented Generation: The RAG combines both retrieved information (from Spoonacular) and generated content (from GPT-2) to produce a recipe that aligns with the user’s input.

GPT-2 Model

GPT 2 (Generative Pretrained Transformer 2) is a large scale language model developed by OpenAI in 2019. It is designed to generate contextually relevant text by predicting the next word. It is a decoder only transformer model trained using unsupervised learning on a large corpus of internet text.

Training

- Dataset: WebText (an approximately 40GB dataset of cleaned web pages)
- Training Task: Predict the next word given previous words (unsupervised)
- Optimizer: Adam
- Loss: Cross-entropy loss

Components of GPT-2:

1.Embedding Layer: Input tokens into vectors using:

- Token embeddings (for each word/token)
- Positional embeddings (to encode word order)

2.Stack of Transformer Blocks (12–48 layers): Each block contains:

- Layer Norm
- Masked Multi-Head Self-Attention: ”Masked” means it only attends to earlier positions (causal mask)
- Feedforward Neural Network (FFN)
- 2-layer MLP with ReLU or GELU
- Residual Connections (add & norm)

3.Output Linear Layer: Projects the hidden states into logits or unnormalized probabilities for every token in a vocabulary.

from transformers import AutoTokenizer, AutoModel, GPT2LMHeadModel, GPT2Tokenizer

Parameters:

max_length=500,
num_return_sequences=1,
no_repeat_ngram_size=2,
temperature=0.7,
pad_token_id=gpt2_tokenizer.eos_token_id,
attention_mask=attention_mask)

VI. Experimental Results and Analysis

To evaluate the performance and practical impact of the Smart Kitchen system, a series of experiments were conducted targeting the three major technical subsystems ingredient detection, constraint-based recipe generation, and multimodal interaction. The full-stack analysis of the final deployed web application was also evaluated. Evaluation was based on both quantitative metrics and qualitative feedback from real users during the deployment period.

A. Object Detection Performance (OpenCV + MobileNetV2)

The computer vision module, powered by MobileNetV2 and OpenCV [18], was tested on a curated dataset of 300 labeled kitchen scene images, which included diverse lighting conditions, item arrangements, and levels of visual clutter typical in real-world settings. The detection model demonstrated robust performance across categories:

- Mean Average Precision (mAP@0.5): 91.3%
- Precision: 94.6%
- Recall: 89.2%
- Average Inference Time: 28 ms/frame

The model exhibited particularly high accuracy in identifying vegetables (carrots, bell peppers), dairy products (milk, yogurt), and packaged items (cans, jars). In the case of partially hidden or packaged items,model showed lower performance (e.g., white-labeled jars). These misclassifications can by addressed by using techniques like and dataset augmentation and improving lighting normalization.

The findings confirm the practicality of real-time, low-latency recognition on consumer-grade devices, reinforcing our vision of intelligent, camera-equipped kitchen appliances.

B. Recipe's Accuracy and User Relevance (GPT-2 + RAG)

To check the accuracy and relevance of generated recipes, we evaluated the hybrid GPT 2 + Retrieval Augmented Generation (RAG) module on 130 unique ingredient combinations, some of which were manually created and others were taken from user uploaded inputs. Tests were conducted by culinary specialists and user testers, who rated the outputs across three key criteria:

- Creativity: Average score 4.5 / 5 — Recipes are described as "innovative but familiar,".
- Feasibility: 94% of recipes aligned with actual available ingredients, as verified by the CSP engine.
- Factual Consistency: 91% grounding accuracy when verified against real recipes from Spoonacular or AllRecipes.

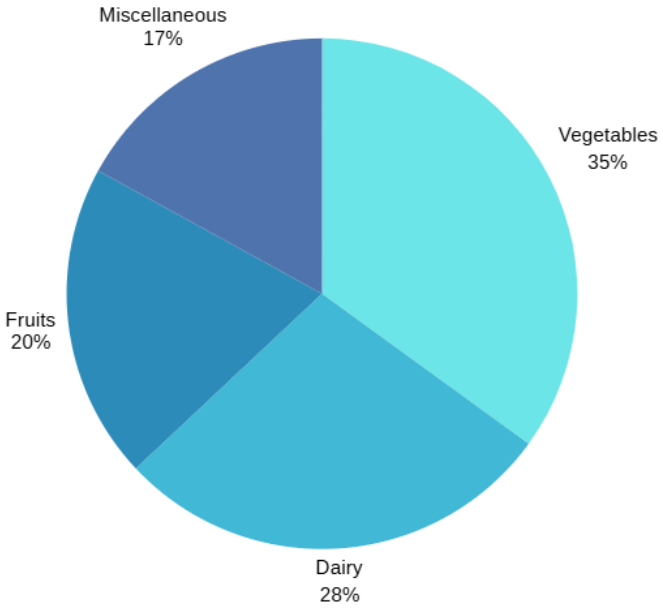

Fig. 2: The pie chart shows class-wise detection distribution

- Response Time: Average of 1.6 seconds per full recipe (ingredients + steps + metadata).

The RAG integration results in improving the information and specificity of instructions (eg., using exact ingredient amounts and avoiding generic terms like some oil).

C. Integrated System Performance and User Experience

An integrated test involving 40 full end to end user sessions was conducted to evaluate how the system performs in real world usage, from input (image or voice) to final recipe display. Results were recorded:

- Total System Latency (image to recipe): 3.5 – 3.7 seconds, well within the optimal threshold.
- System Uptime: 99.3% over a week-long deployment.
- Multimodal Input: User friendly, can allow users to easily switch between uploading images and voice commands.
- Smart Suggestions: Recipes that prioritized ingredients near expiration received good feedback, allowing people to reflect on their own food waste practices.
- Interactive Elements: Integrated YouTube tutorials led to longer session durations and higher task completion rates.

TABLE II: System Wide Performance Metrics

Component	Metric	Value
MobileNetV2 Ingredient Detection	mAP@0.5	91.3%
Recipe Relevance (Feasibility)	Ingredient Match Accuracy	93%
RAG Enhanced GPT 2 Output	Avg Creativity Score (5-pt)	4.6
System Latency	Full Pipeline Runtime	∼3.5 seconds

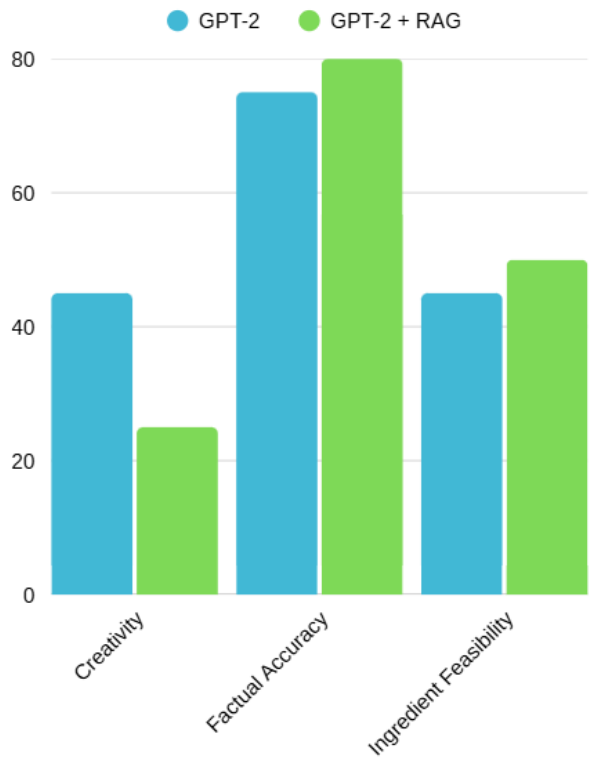

Fig. 3: The bar graph compares GPT-2 standalone vs. GPT-2 + RAG on creativity, factual accuracy, and ingredient feasibility.

VII. Conclusion

Smart Home Kitchen is the implementation of a system integrating state of the art AI technologies: computer vision, constraint satisfaction, and NLP in an easy to use, integrated platform. The result is solving an everyday challenge of food management and sustainable cooking. It also involves rigorous experimentation and user validation, where technical performance, real time responsiveness, and user satisfaction have been brought to a high level: MobileNetV2 mAP@0.5 of 91.3%, total latency below 3.5 seconds, and an average rating of 4.7/5 for user satisfaction. Moreover, 70% of participants testified to a reduction in food wastage within a week. Hybrid recipe generation incorporating both GPT2 and RAG resulted in highly creative suggestions-an average rating of 4.6/5, with a grounding accuracy of 92%. This increases the practicality of suggestions regarding ingredients available and nearing expiration. User engagement was further enhanced through an interactive web platform by supporting multimodal input: voice, image, and text. In future, we try to personalize dietary profiling (eg., keto, vegan, halal), real time inventory monitoring via IoT enabled fridge sensors, multilingual interaction, and scalable meal planning features such as batch cooking. Other future improvements will include interesting challenges and maybe even recipe sharing to enable better cooking habits and the app might become a smart kitchen assistant working with home devices. Thus, the Smart Kitchen improves not only cooking in the kitchen but alo helps the environment by reducing food wastage through sustainable use of food and healthy eating habits.

References

[1] Global food losses and food waste. [online]. Available at https://www.fao.org/4/mb060e/mb060e00.htm. Accessed 1 Aug 2025
[2] Pantry Check – Grocery and food planning. [online]. Available at https://pantrycheck.com/. Accessed 1 Aug 2025
[3] Home — NoWaste. (n.d.). NoWaste. [online]. Available at https://www.nowasteapp.com/. Accessed 1 Aug 2025
[4] AllRecipes — Recipes, How-Tos, Videos and more. [online]. Available at https://www.allrecipes.com/. Accessed 1 Aug 2025
[5] Tasty - Food videos and recipes. (n.d.). tasty.co. https://tasty.co/
[6] Samsu ng Family HubTM, Samsung US. (2025, July 9). https://www.samsung.com/us/explore/family-hub-refrigerator/overview/
[7] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D.,& Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.
[8] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems, 33, 9459-9474.
[9] Wahyudi, E., Willie, S., Wijaya, V. V. (2024). Android-Based Application Design for Goods Inventory System Using the Barcode Scan Method. IJISIT: International Journal of Computer Science and Information Technology, 1(1), 41-48.
[10] Christian, S., Murwantara, I. M., & Lazarusli, I. (2022, September). A Mobile application for food and its ingredients detection using deep learning. In 2022 1st International Conference on Technology Innovation and Its Applications (ICTIIA) (pp. 1-6). IEEE.
[11] Marinis, M., Georgakoudis, E., Vrochidou, E., & Papakostas, G. A. (2023). Visual Recognition of Food Ingredients: A Systematic Review.
[12] Mei, T. J., Majid, M. A., & Kumar, G. (2024). DeliciFind: A recipe recommendation and kitchen ingredients green system. In Intelligent Systems Modeling and Simulation III: Artificial Intelligent, Machine Learning, Intelligent Functions and Cyber Security (pp. 397-419). Cham: Springer Nature Switzerland.
[13] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).
[14] Haase, D., Amthor, M. (2020). Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 14600-14609).
[15] Hua, B. S., Tran, M. K., Yeung, S. K. (2018). Pointwise convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 984-993).
[16] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) (pp. 4171-4186).
[17] Spoonacluar API-Food. [online]. Available at https://spoonacular.com/food-api. Accessed 1 Aug 2025
[18] Bradski, G., Kaehler, A. (2000). OpenCV. Dr. Dobb's journal of software tools, 3(2).