What’s in my Fridge

UML501 Machine Learning Project Report
Semester Evaluation

Submitted by:
(102303864) Krishiv Goyal
(102303867) Bedangi Saha
(102303861) Kangan Arora

BE Third Year, COE/
Group No: 3C62

Submitted to:
Dr Manisha Malik

Computer Science and Engineering Department
TIET, Patiala

INDEX

1.Introduction

Cooking is an essential aspect of everyday life, yet deciding what to prepare often becomes a challenge for individuals and families. While most households have a variety of ingredients available, people frequently struggle to identify suitable recipes using only what they already possess. This uncertainty not only leads to confusion but also results in unnecessary grocery purchases and, in many cases, food wastage due to overlooked or unused ingredients. With the growing integration of artificial intelligence into daily routines, there is a significant opportunity to simplify and optimize the meal-planning process.

To address this problem, our project presents an ranking based Recipe Recommendation System designed to help users make efficient use of the ingredients they have at home. The system allows users to input available ingredients along with their respective quantities. Based on this information, the model recommends recipes, prioritizing those that can be prepared with minimal additional ingredients. This ranking approach helps users plan meals more economically, reduce food waste, and make smarter kitchen decisions.

Enhancing the user experience further, the system includes a Computer Vision–driven Vegetable Recognition Module. Instead of manually typing ingredient names, users can simply capture an image of vegetables, and the system automatically identifies them. Although the current version supports recognition of a limited set of vegetables, it effectively demonstrates the potential of integrating computer vision with AI-driven recipe recommendation. This combination offers a seamless bridge between real-world ingredients and a digital cooking assistant, paving the way for smarter and more intuitive kitchen automation.

2.Need Statement

The increasing complexity of modern lifestyles has made meal planning, ingredient management, and daily cooking more challenging than ever. Several factors highlight the need for an intelligent AI-based Recipe Recommendation System:

1. **Food Wastage as a Global Concern**: A significant portion of household food is discarded due to poor planning or failure to utilize ingredients before they expire.
2. **Time Saved in Meal Planning:** Many users spend considerable time browsing apps, websites, or social media for ideas. The proposed system eliminates this effort by instantly generating recipe options that match available ingredients, making everyday meal planning faster and more efficient.
3. **Reduction in Unnecessary Grocery Shopping:** The system ranks recipe suggestions according to the number of additional ingredients required, making cooking more economical.
4. **Support for Users with Limited Cooking Knowledge:** The system acts as a smart assistant, guiding users with simple and relevant recipe suggestions, thus making cooking more accessible and less intimidating.
5. **Convenience for People with Busy Lifestyles:** Many individuals such as students, professionals, and families lack the time to plan meals or track what ingredients remain at home. By automating recipe discovery and reducing decision fatigue, the system becomes extremely valuable for users with tight schedules.

3.Problem Statement

Many individuals struggle to decide what to cook despite having multiple ingredients at home. Matching available items, often in uncertain quantities, to suitable recipes can be difficult. Users spend considerable time browsing recipe sites, comparing ingredient lists, and checking what they have, leading to confusion, repetitive meals, or food waste.

This problem grows when users cannot correctly identify certain vegetables, especially those with similar appearances. Limited cooking knowledge further restricts their ability to explore diverse recipes, reducing efficient use of available ingredients.

Existing recipe platforms offer only partial solutions. They typically provide generic suggestions without analyzing users' actual ingredients or checking for quantity sufficiency. Most also lack computer vision features, requiring manual ingredient entry, which is time-consuming and error-prone.

To overcome these gaps, there is a need for an intelligent system combining ingredient-based recipe recommendation with automated vegetable recognition. By allowing users to enter ingredient quantities and upload images for identification, such a system can recommend recipes ranked by the number of additional ingredients required, reduce decision-making time, minimize unnecessary shopping, and help prevent food wastage.

4.Objectives

1. To develop a ranking based recipe recommendation system that analyzes user-provided ingredients to determine which recipes can be prepared using the available items.
2. To minimize dependency on additional ingredients by ranking recipes in an optimized order, prioritizing dishes that require the fewest new items.
3. To integrate a computer vision–based vegetable recognition module that allows users to identify vegetables by simply capturing or uploading an image.
4. To reduce household food wastage by ensuring efficient utilization of ingredients already present in the user's kitchen.
5. To simplify and accelerate meal planning through fast, accurate, and intelligent recipe suggestions without the need for manual searching or comparison.
6. To create a user-friendly platform that enables users to manage ingredients, recognize vegetables, and explore suitable recipes easily, even with minimal cooking experience.

5.Existing Approach

- **Traditional Recipe Websites and Apps (e.g., AllRecipes, Tasty, Sanjeev Kapoor, Hebbars Kitchen)**
  - These platforms provide extensive collections of recipes but do not consider the specific ingredients that a user currently has at home. Users must manually search and filter through numerous recipes, which is time-consuming and often inefficient. Moreover, these platforms do not optimize recipe suggestions to minimize the need for additional ingredients.

- **Ingredient-Based Search Applications**
  - Some applications allow users to search for recipes based on ingredients. However, they typically rely on manual entry of each ingredient. These apps lack mechanisms to calculate or rank recipes based on the number of extra ingredients required.

- **Standalone Image Recognition Models (Vegetable/Fruit Detection)**
  - Various machine learning and deep learning models exist for detecting vegetables or fruits from images. However, these models operate independently and are not integrated with recipe recommendation systems. They generally support only a limited set of categories and cannot assist with quantity-based recipe planning or meal preparation
- Grocery and Meal-Planning Applications
  - Several apps help users plan meals, track groceries, or manage kitchen inventories. While useful, these systems do not combine computer vision with recipe optimization. They may suggest meals but do not dynamically match them with available ingredient quantities or prioritize recipes that require minimal additional purchases.

.

6.Dataset Description

Dataset Link : https://data.mendeley.com/datasets/xsphgmmh7b/1

Dataset Name: IndianFoodDatasetCSV.csv

Dataset Size

- Number of rows (instances): ~ 6000+ recipes
- Number of columns (features): 13

Data Nature & Content Properties

- Contains Unstructured Text Data (ingredients + instructions).
- Contains Categorical Variables (cuisine, course, diet).
- Contains Numeric Variables (prep, cook, total time, servings).
- Some recipes appear originally in Hindi; the dataset includes translated versions to maintain consistency.
- Data scraped from an external cooking website, so may include:
    - Inconsistent formatting
    - Typos or translation imperfections
    - Irregular ingredient formatting

Extra field added in dataset

- From the website URL field, the dish image URLs were web-scraped and added to the dataset, enabling visual display of recipes and supporting image-based recommendation interfaces.

7.Methodology

1. Ingredient Detection Methodology

The first phase aims to identify the vegetables present in an input image. This is achieved through a custom-trained object detection model based on the YOLOv8 architecture.

1.1 Data-Driven Detection Approach

- A custom dataset containing multiple vegetable categories was curated and annotated to train the detection model. The dataset includes diverse samples of each vegetable class, captured under different lighting conditions, orientations, and backgrounds to ensure robustness. The model learns visual patterns associated with each vegetable category during training.

1.2 YOLOv8 as the Detection Framework

- YOLOv8 was selected due to its proven ability to perform accurate detection in real time. The model processes an image holistically and predicts bounding boxes along with class probabilities for all visible vegetables. Its multi-scale feature extraction allows the system to detect vegetables of varying sizes and shapes effectively.

1.3 Recognition Output

- After inference, the model provides a list of recognized vegetables present in the image. These detected ingredients become the core input for the next phase, where they are matched with suitable recipes. The detection process ensures that downstream recommendations are grounded in the actual contents of the user-provided image.

2. Dish Recommendation Methodology

- Once the vegetables are identified, the system determines which dishes can be prepared using those ingredients. This is accomplished by analyzing and comparing textual recipe information.

2.1 Recipe Textual Analysis

- The recipe dataset contains detailed ingredient lists for thousands of Indian dishes. To extract meaningful information from this unstructured text, the ingredient descriptions are cleaned, normalized, and transformed into a standard format. This ensures consistency across recipes and enables reliable comparison.

2.2 Recommendation Strategy

- Instead of relying on vector similarity measures, the system directly compares the user's detected ingredients with each recipe's ingredient list. For every recipe, the overlap with the user-provided ingredients is calculated, and the number of additional ingredients required is determined. Recipes that require fewer extra ingredients are ranked higher, ensuring that users are recommended dishes they can prepare more easily with what they already have.

2.3 Output Generation

- The top-ranked dishes are selected and presented as recommendations. These recipes best match the combination of vegetables detected in the input image, enabling the user to discover meals they can prepare with ingredients already available to them.

3. Integrated System Workflow

- The overall workflow follows a structured pipeline:
- Input: User provides an image containing vegetables.
- Detection: YOLOv8 identifies all visible vegetable classes.
- Text Processing: Recipe dataset ingredients are normalized and vectorized.
- Recommendation Strategy: Suggestions ranked based on least number of extra ingredients required.
- Recommendation: Highest-ranking dishes are suggested to the user.
- This methodology ensures the system is both data-driven and user-centered, combining vision-based detection with semantic text analysis to deliver meaningful recipe recommendations.

8.Implementation Details

The system is implemented in two major components:
 (1) Vegetable Detection using YOLOv8 and
 (2) Recipe Recommendation.
 Both modules are integrated to produce dish suggestions based on detected vegetables from an image.

1.1 **Vegetable Dataset Preparation**

1.2.1 **Dataset Structure**

A custom dataset containing 15 vegetable classes was collected and arranged into folders. Each folder contains 50–1000 images of the corresponding vegetable in various lighting, angles, and backgrounds.

1.3 Annotation

Each image was annotated using LabelImg or Roboflow.
YOLO Format of Labels:
Each annotation is saved as a text file:
<class_id> <x_center> <y_center> <width> <height>
All annotations were stored inside a labels/ directory matching the structure of images/.
1.4 Train–Validation Split: 80% for training and 20% for validation

1.5 YOLOv8 Model Training
1.5.1 Data Configuration File (data.yaml)
A configuration file was created with 15 different fruits and vegetables.
1.5.2 Training Script

```
from ultralytics import YOLO
model = YOLO("yolov8s.pt")
model.train( data="data.yaml", epochs=80, imgsz=640, batch=16, device=0)
```

1.5.3 Training Outputs
YOLO automatically generates:

- runs/detect/train/weights/best.pt
- runs/detect/train/weights/last.pt
- mAP curves
- Loss curves (objectness, classification, bounding box loss)

1.6 Vegetable Detection (Inference Phase)

After training, the best.pt model is used for prediction.

Inference Code

```
model = YOLO("best.pt")
results = model.predict("input_image.jpg", conf=0.45)
```

Extraction of Detected Vegetables

1.7 Recipe Dataset Processing

Cleaning Steps

```
df = df.drop(columns=['Srno', 'Image_URL'])
df = df.drop_duplicates(subset=['RecipeName'])
df = df.dropna(subset=['Ingredients'])
```

Text Preprocessing

```
text = str(text).lower()
text = re.sub(r'[^\x00-\x7F]+',' ', text)
text = re.sub(r'[^a-z, ]',' ', text)
```

1.8 Ingredient Normalization and Preprocessing

Normalization includes lowercasing, trimming spaces, and applying a custom normalization function to merge synonymous terms.

```
set([normalize_ingredient_name(i) for i in user_ingredients])
```

Each recipe undergoes the same cleaning process

```
[normalize_ingredient_name(i) for i in row["CleanedIngredients"].split(",")]
```

1.9 Matching Logic and Overlap Computation

Number of ingredients the user already has that are present in the recipe.

```
intersection = len(user_set & recipe_set)
```

Recipes with zero overlap are discarded.

1.10 Extra Ingredient Calculation

To prioritize recipes that can be prepared easily, the system calculates how many additional ingredients the recipe requires.

1.11 Final Ranking Mechanism

Recipes are ranked using two criteria:

- Primary criterion: Fewer extra ingredients required.
- Secondary criterion: Higher overlap with the user's available ingredients.

This ensures:

- Recipes needing 0–2 additional ingredients appear at the top.
- Among recipes requiring the same number of extra items, those with more matched ingredients are ranked higher.

9.Results

a. YOLOv8 Detection Model

The YOLOv8 model processes the input image and outputs:

- Detected vegetables present in the frame
- Bounding boxes marking their positions
- Confidence scores indicating detection reliability
- Resulting arrray indicating all the detected ingredients

These results form the basis for determining which ingredients are available for recipe matching.

b. Recipe Recommendation Module

Based on the detected ingredients, the recommendation module generates:

- A ranked list of recipes
- Ranking based on ingredient overlap and minimum additional items required

This ensures that recipes using the maximum available ingredients and requiring the fewest extra items appear first.

10.Evaluation

a. Detection Model (YOLOv8)

To assess the performance of the vegetable recognition system, the following metrics were used:

1. Measures how many of the detected vegetables were correctly classified.
2. Measures how many of the actual vegetables present in the image were successfully detected.
3. mAP@70.9 (Mean Average Precision at 70.9%)
   a. Evaluates the accuracy of detections with a 50% Intersection-over-Union threshold.
4. mAP@50–95 for the trained model showing overfitting.

b. Recommendation Module Evaluation

1. Ingredient Overlap Score

- This metric measures how many of the user's detected ingredients are already present in a recipe.
- A higher overlap score indicates that the recipe makes good use of the available ingredients.
- Recipes with zero overlap are excluded from recommendations.

2. Extra Ingredients Count

- This metric reflects how many additional ingredients are required for the user to prepare the recipe.
- Fewer extra ingredients indicate a more accessible recipe.
- This metric is the primary factor in ranking.

3. Final Ranking Order

- Recipes are sorted first by Extra Ingredients Count (ascending).
- If two or more recipes require the same number of extra ingredients, they are further ranked by Ingredient Overlap Score (descending).

yolov8v Model

Metric	Value
Precision (P)	0.71
Recall (R)	0.63

Trained Model

```
CLASSIFICATION REPORT:
              precision    recall  f1-score

        Bean       1.00      1.00      1.00
Bitter_Gourd       1.00      1.00      1.00
Bottle_Gourd       1.00      1.00      1.00
     Brinjal       1.00      1.00      1.00
    Broccoli       1.00      1.00      1.00
     Cabbage       1.00      1.00      1.00
    Capsicum       1.00      1.00      1.00
      Carrot       1.00      1.00      1.00
 Cauliflower       1.00      1.00      1.00
    Cucumber       1.00      1.00      1.00
      Papaya       1.00      0.99      1.00
      Potato       1.00      1.00      1.00
     Pumpkin       1.00      1.00      1.00
      Radish       1.00      1.00      1.00
      Tomato       1.00      1.00      1.00

    accuracy                           1.00
   macro avg       1.00      1.00      1.00
weighted avg       1.00      1.00      1.00
```

11.Conclusion

The recipe recommendation system developed in this project addresses a common challenge: deciding what to cook with the ingredients available at home. Many users struggle with meal planning, often wasting ingredients, repeating dishes, or buying unnecessary items. By analyzing the user's ingredients and their quantities, the system suggests recipes that require the fewest additional items, reducing effort, cost, and food wastage.

A major strength of the system is its image-based vegetable recognition module. Users can simply take a picture of vegetables, and the model identifies them automatically, minimizing manual input. Although the current version supports a limited set of vegetables, it makes the system more accessible and demonstrates how computer vision can assist users in real-world cooking decisions.

The project effectively combines ingredient-matching logic, user input processing, and image recognition to build a practical kitchen assistant. By ranking recipes based on additional ingredient requirements, it helps users achieve better variety, reduce waste, and enjoy a more efficient cooking process, especially useful for those with busy schedules or limited cooking experience.

12.Future Works and Limitations

Future Works

1. **Expand the image-recognition dataset** to include more vegetables, fruits, packaged items, and spices for broader ingredient detection.
2. **Integrate nutritional analysis** to recommend healthier recipes based on calories, macronutrients, or specific diet plans.
3. **Introduce user preference learning** so the system can personalize recommendations based on taste, cuisine type, cooking frequency, and feedback.
4. **Add filters for recipe selection,** such as preparation time, difficulty level, cost of missing ingredients, and cooking method.
5. **Enable automatic shopping list** creation based on missing ingredients and quantity requirements.
6. **Enable recipe rating** and feedback so the system can improve future recommendations.

Limitations

1. **Limited Vegetable Recognition Coverage**: The computer vision module is trained on only a small set of commonly used vegetables.
2. **Uneven Recognition Accuracy Across Vegetables**: The CNN model demonstrates high accuracy when identifying certain ingredients and less accuracy for others.
3. **Recipe Dataset Limitations:** The diversity and quality of recipe recommendations depend entirely on the dataset provided.
4. **Lack of Personalization or Adaptive Learning**: The system does not learn from user preferences or historical behavior.