

# FoodLens AI: An AI-Powered Food Inventory and Recommendation System Using YOLOv8-Based Food Detection

Ms. Bhoomika R<sup>1</sup>, Ms. Tejaswini K R<sup>2</sup>, Ms. Vindya A P<sup>3</sup>, Ms. Thanusha J<sup>4</sup>, Prof. Amulya<sup>5</sup>  
<sup>1,2,3,4,5</sup>*Department of Computer Science and Engineering, Shree Devi Institute of Technology*

**Abstract**—Keeping track of what is inside a household refrigerator is still largely a manual task, which makes it hard to know which items are available and how to use them. This paper presents FoodLens AI, a web-based food inventory and recommendation system that uses YOLOv8 object detection to identify food items in refrigerator images and turn the detections into inventory records and recipe suggestions. The system is organized as a React web client that submits images to a FastAPI backend, where a YOLO model performs inference and a PostgreSQL database stores users, inventory, food items, recipes, and recommendation information. A first-generation detector (V1, YOLOv8n) reached an approximate validation precision of 0.825, recall of 0.752, mAP@0.5 of 0.813, and mAP@0.5:0.95 of 0.574, but testing on real crowded refrigerators exposed missed detections and weak generalization. To address this, a second-generation dataset (V2.1) was prepared for a YOLOv8n-seg model by merging the original FoodLens data with a crowded refrigerator dataset over 30 target classes, giving 9,466 images and 53,349 instances after SHA-256 duplicate removal, and passing the implemented validation checks without reported errors. Training of V2.1 has not been completed, so no V2.1 model performance is reported; comparing V2.1 with V1 on crowded scenes is the immediate next step. The contribution is an integrated, system-oriented workflow rather than a new detection algorithm.

**Index Terms**—Food inventory management, YOLOv8, object detection, instance segmentation, smart refrigerator, recipe recommendation, FastAPI.

## I. INTRODUCTION

Household food waste is linked to people's awareness of refrigerator contents, leading to items being forgotten and not tracked properly, which contributes to avoidable waste. Research has focused on smart refrigerator and pantry systems to help improve

inventory management and reduce this issue. Computer vision facilitates the automation of item detection by allowing users to photograph their refrigerators instead of manually entering data. YOLO family detectors, such as YOLOv5, YOLOv7, and YOLOv8, effectively localize and classify food items in a single pass. These detectors are utilized in various studies for tasks such as food categorization, household food recognition, and food-damage detection. Additionally, related research encompasses hardware-assisted refrigerator monitoring, shelf-life prediction, grocery management, and recipe recommendation. These research directions are often addressed as separate components in existing systems. In addition, a detector that performs well on a curated dataset does not automatically work in a real refrigerator, where items overlap, sit behind one another, and are partly hidden. This was observed during the development of FoodLens AI: a first-generation YOLOv8n model gave reasonable validation metrics but missed items and generalized poorly when tested on crowded refrigerators. This paper introduces FoodLens AI, a web-based food inventory and recommendation system leveraging a YOLOv8 detection pipeline. It emphasizes a system-oriented approach, presenting an iterative development process rather than proposing a new detection algorithm. The paper includes four main components: (1) an overview of the architecture linking a React client to a FastAPI service, YOLO inference, and a PostgreSQL database; (2) validation results from the initial version (V1) of the detector; (3) details on the construction and validation of the second-generation dataset (V2.1) aimed at addressing crowded scenes; and (4) a clarification of the results that are currently available versus those that are still pending.

## II. RELATED WORK

### A. Smart Refrigerator and IoT-Based Systems

IoT-based refrigerators integrate sensing technology with cloud and mobile interfaces. The first system utilizes a Raspberry Pi coupled with a Pi Camera, NodeMCU, load cells, and Blynk connectivity, employing a YOLO detector trained on 1,022 images annotated by Roboflow across 14 produce categories. This system also includes capabilities for weight monitoring and recipe retrieval based on recognized items. The second system, described sensor oriented, incorporates an Arduino Uno, DS18B20 temperature sensor, HX711 load cells, MQ-2 gas sensor, ESP8266 Wi-Fi, and ThingSpeak, which delivers low-stock and spoilage alerts using threshold logic, but lacks computer vision functionality. Lastly, FridgeSense features a three-tier architecture, utilizing a Raspberry Pi 4 in the hardware layer, a processing layer employing YOLOv8, OCR, and BiLSTM for parsing expiry dates, and a Firebase-based application layer that includes a recommendation engine and a mobile application. FoodLens AI offers a software-based solution for automatic monitoring of appliances, differentiating itself by utilizing images submitted through a web interface instead of relying on dedicated sensing hardware like cameras, sensors, and microcontrollers. This approach highlights the advantages of automatic monitoring while reducing the dependence on physical modifications of the appliance.

### B. Food Detection and Recognition

Inventory research focuses on aspects such as availability, stock levels, expiry, and waste reduction. Notable tools include FridgeSense, which utilizes OCR for expiry-date reading, and SmartPantry Manager that predicts shelf life using a stacking regressor, achieving an  $R^2$  of 0.9603, RMSE of 8.68 days, and MAE of 4.16 days. SmartPantry has limitations in predicting highly

perishable items and suggests computer vision for item detection. In the grocery sector, a study compares SVM, Random Forest, and Decision Tree classifiers for inventory classification, identifying challenges like real-time personalization, scalability, privacy, and integration of inventory and purchasing. In recipe recommendations, RecipeMate ranks approximately 26,000 to 28,000 Yummly-derived recipes using TF-IDF and cosine similarity based on available ingredients. Other approaches include CNN-based ingredient image recognition for recipe suggestion, with FridgeSense reporting a 73% top-3 recipe selection rate and an average user satisfaction score of 4.6 out of 5. A summarizing table in the document outlines these studies.

### C. Inventory, Expiry, and Recipe Recommendation

Inventory research focuses on aspects such as availability, stock levels, expiry, and waste reduction. Notable tools include FridgeSense, which utilizes OCR for expiry-date reading, and SmartPantry Manager that predicts shelf life using a stacking regressor, achieving an  $R^2$  of 0.9603, RMSE of 8.68 days, and MAE of 4.16 days. SmartPantry has limitations in predicting highly perishable items and suggests computer vision for item detection. In the grocery sector, a study compares SVM, Random Forest, and Decision Tree classifiers for inventory classification, identifying challenges like real-time personalization, scalability, privacy, and integration of inventory and purchasing. In recipe recommendations, RecipeMate ranks approximately 26,000 to 28,000 Yummly-derived recipes using TF-IDF and cosine similarity based on available ingredients. Other approaches include CNN-based ingredient image recognition for recipe suggestion, with FridgeSense reporting a 73% top-3 recipe selection rate and an average user satisfaction score of 4.6 out of 5. A summarizing table in the document outlines these studies.

TABLE I. SUMMARY OF REVIEWED STUDIES

| Ref. | Focus                            | Method                        | Reported result              |
|------|----------------------------------|-------------------------------|------------------------------|
| [1]  | Food tracking, expiry, recipes   | YOLOv8, OCR, Bi-LSTM          | mAP 92.3%; OCR 88.1%         |
| [2]  | Fridge inventory, camera + IoT   | Raspberry Pi, YOLO, load cell | Class accuracy 0.5–0.9       |
| [3]  | Sensor-based inventory, spoilage | Arduino, load cell, MQ-2      | Scenario demonstrations      |
| [4]  | Food-damage detection            | YOLOv8                        | Curves only; no final figure |

|      |                                |                           |                                     |
|------|--------------------------------|---------------------------|-------------------------------------|
| [5]  | Expiry prediction              | Stacking regressor        | $R^2$ 0.9603; MAE 4.16 days         |
| [6]  | Recipe recommendation          | TF-IDF, cosine similarity | Cuisine corrects in 5/5 runs        |
| [7]  | Recipes from ingredient images | CNN, OpenCV               | Accuracy not stated numerically     |
| [8]  | Grocery inventory              | SVM, RF, DT, proposed     | SVM 84.2%; RF 87.5%; proposed 92.1% |
| [9]  | Food type and state            | DenseNet121, cascaded     | State 87%; type 71.35%              |
| [10] | Food detection                 | YOLOv5, YOLOv7            | mAP@0.5: 0.89 vs 0.669              |

#### D. Gap Addressed by the Proposed System

The reviewed studies cover detection, hardware monitoring, expiry prediction, and recipe recommendation, but mostly as separate pieces, and the detection studies are evaluated on datasets far from a crowded household refrigerator. The opportunity addressed here is a softwarebased workflow that connects image-based food detection to persistent inventory records and recipe suggestions, and that treats generalization to crowded scenes as a first-class development concern. FoodLens AI is therefore best understood as an integrative system contribution and not as a replacement for established food-detection research.

### III. PROPOSED SYSTEM

#### A. Design Objectives

FoodLens AI was designed around five objectives: (1) identify food items automatically from refrigerator images; (2) convert detections into structured inventory records; (3) store user, inventory, and recipe data persistently; (4) suggest recipes from the available inventory; and (5) improve robustness to crowded, overlapping items through explicit model versions that can be compared. The web client interacts with the model and the database only through the backend API.

#### B. System Architecture

Fig. 1 illustrates the architecture of a system designed for refrigerator scanning. The system's front end utilizes a React web client, which communicates with the backend via the POST /api/scan endpoint to send images. The backend, implemented using FastAPI, processes these images through YOLO model inference to identify food items, each associated with a class label, location, and confidence score. This extracted data is subsequently integrated into inventory and recommendation algorithms.

Additionally, a PostgreSQL database serves as a persistence layer, storing information about users,

inventories, food items, recipes, and recommendations, ensuring that the inventory state is maintained across multiple scans.

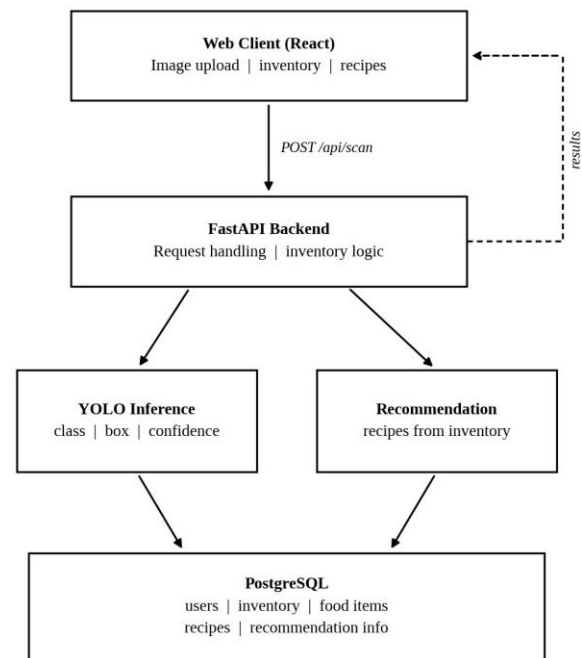


Fig. 1. High-level architecture of the FoodLens AI system.

#### C. Model Versions

This work employs two model generations for object detection and segmentation in culinary applications. The first model, V1, is a YOLOv8n object detector that acts as the baseline. Despite showing promising validation metrics (referenced in Section V), testing on real crowded refrigerator images revealed issues such as missed detections and poor generalization. The second model, V2.1, utilizes a YOLOv8n-seg instance segmentation approach, specifically designed for 30 FoodLens classes.

It is to be trained on a unified dataset combining the original FoodLens data with additional crowded-refrigerator images. The key research question investigates whether V2.1 demonstrates superior

generalization compared to V1 when exposed to the complexities of real-world refrigerator images. As of the current writing, the dataset for V2.1 has been prepared and validated; however, actual training has yet to take place, categorizing V2.1 as a prepared but untrained model.

#### D. V2.1 Dataset Construction

The V2.1 dataset was created through a four-step process: first, merging the original FoodLens dataset with a crowdedrefrigerator dataset and aligning them to 30 FoodLens target classes. Second, duplicate images were eliminated by comparing their SHA-256 hashes. Third, annotations were standardized for segmentation, retaining native polygon annotations as segmentation masks and converting bounding-box annotations into box-derived pseudo-masks. The completed dataset comprises 9,466 images and 53,349 instances, including 2,705 native segmentation labels and 50,644 pseudo-masks.

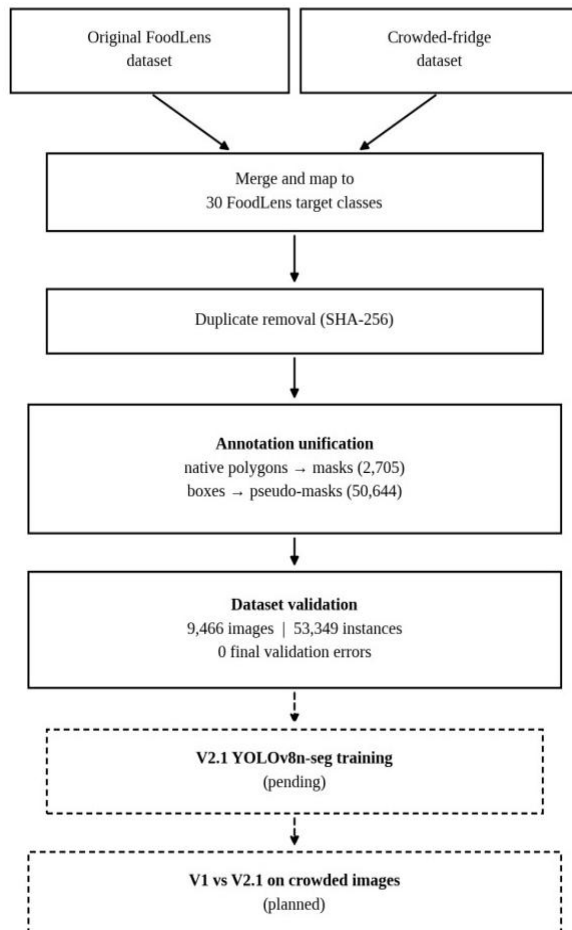


Fig. 2. Construction and validation of the V2.1 dataset; dashed boxes mark steps not yet completed.

## IV. IMPLEMENTATION

### A. Frontend

The client is a React web application. It submits refrigerator images to the backend through the POST `/api/scan` endpoint and presents the returned information to the user.

### B. Backend and Inference

The backend is implemented with FastAPI. On each scan request it loads the image, runs YOLO model inference, and passes the detections to the inventory and recommendation logic. For the V1 detector, each detection consists of a class label, a bounding box, and a confidence score. For the planned V2.1 segmentation model, each detected instance is intended to produce a segmentation mask in addition to these outputs.

### C. Persistence

A PostgreSQL database serves as the persistence layer, storing critical data such as users, inventory, food items, recipes, and recommendations. This enables the reuse of results from scans. Additionally, Table II outlines the technology stack employed.

TABLE II. TECHNOLOGY STACK OF FOODLENS AI

| Layer            | Technology  | Role                                                              |
|------------------|-------------|-------------------------------------------------------------------|
| Front end        | React       | Image submission and display of results                           |
| API              | FastAPI     | POST <code>/api/scan</code> ; request handling and orchestration  |
| Detection (V1)   | YOLOv8n     | Object detection (baseline)                                       |
| Detection (V2.1) | YOLOv8n-seg | Planned instancesegmentation model, 30 classes                    |
| Database         | PostgreSQL  | Users, inventory, food items, recipes, recommendation information |

## V. EVALUATION AND DISCUSSION

FoodLens AI is a prototype, and the evaluation focuses on specific aspects: the validation metrics of version 1 (V1), the composition and validation of the version 2.1

(V2.1) dataset, and qualitative observations from real-world testing. However, no metrics are available for V2.1 model performance, inventory quantity accuracy, or recommendation quality, as these have not been measured.

#### A. Evaluation Metrics

Detection quality is evaluated using Intersection over Union (IoU), which is defined as the area of overlap between the predicted region (Bp) and the ground-truth region (Bgt), calculated as  $\text{IoU} = \text{Area}(\text{Bp} \cap \text{Bgt}) / \text{Area}(\text{Bp} \cup \text{Bgt})$ . Precision (P) is determined by the ratio of true positives (TP) to the sum of true and false positives (TP + FP), while recall (R) measures the ratio of true positives to the sum of true positives and false negatives (TP + FN). Average precision (AP) is derived from the area under the precision-recall curve for each class, with mean average precision (mAP) representing the average of AP across N classes. The metric mAP@0.5 employs an IoU threshold of 0.5, whereas mAP@0.5:0.95 averages IoU thresholds between 0.5 and 0.95 in increments of 0.05 to emphasize tighter localization.

#### B. V1 Validation Results

Table III provides the validation metrics for the V1 YOLOv8n detector. It demonstrates a precision of 0.825, suggesting a relatively low rate of false-positive detections.

The recall of 0.752 indicates that some actual instances were missed during validation. Additionally, the discrepancy between mAP@0.5 (0.813) and mAP@0.5:0.95 (0.574) reveals that the quality of localization diminishes as the Intersection over Union (IoU) criteria become more stringent.

TABLE III. APPROXIMATE VALIDATION METRICS OF V1 (YOLOV8N)

| Metric       | Value |
|--------------|-------|
| Precision    | 0.825 |
| Recall       | 0.752 |
| mAP@0.5      | 0.813 |
| mAP@0.5:0.95 | 0.574 |

#### C. Real-World Crowded Refrigerator Testing

When tested on crowded real-world refrigerator images, V1 exhibited missed detections and

generalization issues. These findings, although qualitative, serve as the primary motivation for the development of the V2.1 dataset, although no aggregate or per-class figures from the testing are provided.

#### D. V2.1 Dataset Validation

Table IV provides a summary of the V2.1 dataset, revealing that approximately 95% (50,644 out of 53,349) of the instances consist of pseudo-masks, which are created from bounding boxes. In contrast, only 2,705 instances, representing about 5%, feature native polygon annotations.

Consequently, the quality of the masks is constrained by the original bounding box accuracy. While segmentation output is anticipated to yield more precise extents for overlapping items than the boxes alone, this hypothesis has not been empirically validated to date.

TABLE IV. COMPOSITION OF THE V2.1 DATASET

| Item                       | Value   |
|----------------------------|---------|
| Target classes             | 30      |
| Images                     | 9,466   |
| Instances                  | 53,349  |
| Native segmentation labels | 2,705   |
| Pseudo-masks from boxes    | 50,644  |
| Duplicate removal          | SHA-256 |
| Final validation errors    | 0       |
| V2.1 training status       | Pending |

#### E. Discussion

The V1 mAP@0.5 of 0.813 is less than the YOLOv5 score of 0.89 and FridgeSense's 92.3%, but direct comparisons are not valid due to differing classes and datasets used in those studies. FoodLens AI's primary strength is its integration, combining detection with continuous inventory and recipe data in a seamless web workflow.

It recognizes the disparity between validation metrics and real-world performance, addressing it through future model revisions, without claiming to have developed a new detection or recommendation algorithm.

## VI. LIMITATIONS

Several limitations of the study are acknowledged: Firstly, V2.1 has not been trained, leaving the question of its generalization compared to V1 unanswered. Secondly, V1 figures are based on validation-set data, lacking independent testing on held-out crowded images, with no per-class results provided. Thirdly, many V2.1 masks are pseudomasks derived from boxes. Fourthly, the V2.1 detector aims to recognize only 30 target classes, meaning food items outside these classes will not be detected. Fifthly, aspects such as quantity estimation, expiry-date reading or prediction, and low-stock alerts have not been addressed in this paper, even though they are mentioned in the literature. Sixthly, the quality of recipe recommendations has not been assessed with users or against a benchmark. Lastly, neither inference latency nor deployment performance has been measured.

## VII. FUTURE WORK

The next steps involve training version 2.1 (V2.1) and comparing its performance against version 1 (V1) using an independent set of crowded refrigerator images that were not included in the training process. Performance metrics like box and mask precision, recall, and mean Average Precision (mAP), alongside per-class Average Precision (AP), will be reported. For classes where mask quality is crucial, manually annotated polygons may be used instead of pseudo-masks. Additionally, enhancements to the inventory layer will include management of quantities, handling duplicate objects, and monitoring expiry with lowstock alerts, as referenced in sources [1], [3], and [5]. Finally, a user study will evaluate recipe recommendations, and inference latency will be assessed in the intended deployment environment.

## VIII. CONCLUSION

This paper presents FoodLens AI, a web-based food inventory and recommendation system featuring a React client, FastAPI backend, YOLOv8 inference, and PostgreSQL database. The initial YOLOv8n detector reported validation metrics of 0.825 precision, 0.752 recall, 0.813 mAP@0.5, and 0.574 mAP@0.5:0.95, but struggled with missed detections

in crowded environments. To improve this, a V2.1 segmentation dataset of 9,466 images and 53,349 instances was created and successfully validated. Future work will focus on training with this dataset to evaluate its generalization against V1. Key contributions include the system architecture, detailed dataset construction documentation, and a realistic performance assessment.

## ACKNOWLEDGMENT

The authors express gratitude to the Department of Computer Science and Engineering at Shree Devi Institute of Technology for their support of this work, which constituted a major project.

## REFERENCES

- [1] S. Vadlamudi, K. S. Shashikala, S. D. S. Ramesh, R. Calvin Francis, and G. Rani E, "FridgeSense: An AI-Powered Solution for Smart Food Tracking and Recipe Planning," in *2025 IEEE 7th Int. Conf. Computing, Communication and Automation (ICCCA)*, 2025, doi: 10.1109/ICCCA66364.2025.11325426.
- [2] K. B. D. K. Yadav, O. S. Dalvi, D. S. Bin, and R. N, "IoT Based Smart Refrigerator for Food Management System," in *2024 Int. Conf. Emerging Research in Computational Science (ICERCS)*, 2024, doi: 10.1109/ICERCS63125.2024.10895890.
- [3] B. J. Nandini, M. L. Sahithi, S. G. Harshika, T. S, and G. M, "Smart Refrigerator for Enhancing User Convenience and Food Management," in *2024 Second Int. Conf. Networks, Multimedia and Information Technology (NMITCON)*, 2024, doi: 10.1109/NMITCON62075.2024.10699055.
- [4] A. Dhelia, S. Chordia, and K. B, "YOLO-based Food Damage Detection: An Automated Approach for Quality Control in Food Industry," in *2024 8th Int. Conf. I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, 2024, doi: 10.1109/I-SMAC61858.2024.10714664.
- [5] P. N. B. Sree, R. M. Hegde, Sahana, and S. N, "SmartPantry Manager: An Intelligent System for Grocery Management and Food Waste Reduction," in *2025 9th Int. Conf. Computational System and Information Technology for*

- Sustainable Solutions (CSITSS), 2025, doi: 10.1109/CSITSS67709.2025.11295133.
- [6] A. Padmavathi and D. Sarker, "RecipeMate: A Food Media Recommendation System Based on Regional Raw Ingredients," in 2023 14th Int. Conf. Computing Communication and Networking Technologies (ICCCNT), 2023, doi: 10.1109/ICCCNT56998.2023.10307728.
- [7] A. E, M. M. Mano, S. Nandhini, and H. K. Palani, "Recipe Recommendation using Image Classification," in 2023 5th Int. Conf. Inventive Research in Computing Applications (ICIRCA), 2023, doi: 10.1109/ICIRCA57980.2023.10220811.
- [8] M. Jagatheesan, R. Karthick, V. G. Yokesh, P. DineshKumar, K. Saravanan, and K. Geetha, "Intelligent Grocery Management System Using IoT for Personalized Shopping and Inventory Management," in 2025 Int. Conf. Advanced Computing Technologies (ICoACT), 2025, doi: 10.1109/ICOACT63339.2025.11005200.
- [9] S. S. Alahmari and T. Salem, "Food State Recognition Using Deep Learning," IEEE Access, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3228701.
- [10] S. S, P. Kokil, A. R, and N. V. Sai Manoj, "Enhanced Food Classification System Using YOLO Models for Object Detection Algorithm," in 2023 14th Int. Conf. Computing Communication and Networking Technologies (ICCCNT), 2023, doi: 10.1109/ICCCNT56998.2023.10307943.
- [11] J. Redmon, S. Divvala, R. Girshick, and F. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.