Available *online* at : http://jitce.fti.unand.ac.id/

JITCE (Journal of Information Technology and Computer Engineering)

| ISSN (Online) 2599-1663 |

Research Paper

Food Classification and Monitoring System in Refrigerators Using YOLO Algorithm

Laellatul Husna [1], Rian Ferdian [1], Yoan Purbolingga [1]

[1]*Department of Computer Engineering, Faculty of Information Technology, Universitas Andalas, Padang City, West Sumatra, Indonesia*

ARTICLE INFORMATION

Received: August 27th, 2025
Revised: November 29th, 2025
Available online: December 30th, 2025

KEYWORDS

YOLO, Object Detection, Food Monitoring, Refrigerator, Expiration

CORRESPONDENCE

Phone: -

E-mail: laellatulhusna@gmail.com

A B S T R A C T

Monitoring food freshness in refrigerators remains a challenge for many users, often leading to food spoilage and waste due to the absence of an automatic monitoring system. This study proposes a computer vision–based food. A monitoring system that leverages the YOLOv5 algorithm to automatically detect and categorize food items through camera input and deliver real-time notifications to users via a connected application. Experimental results demonstrate that YOLOv5 achieves an average accuracy of over 90% across various distances and object positions. Despite challenges related to limited datasets and lighting variations inside the refrigerator, the system offers a practical and innovative solution to reduce food spoilage, minimize household food waste, and support more efficient food storage management.

INTRODUCTION

Refrigerators are essential household appliances that play a vital role in preserving food freshness and extending shelf life. However, many users still face difficulties in monitoring food conditions, which often results in food spoilage and unnecessary waste. According to the *Food and Agriculture Organization (FAO)*, food waste remains a global concern, with households being among the largest contributors due to delayed recognition of stored food conditions [1].

Several methods have been introduced to address food waste, including manual labeling and recording, as well as the use of smart refrigerator technologies. Nevertheless, most existing solutions still rely heavily on direct user intervention and are not fully integrated into automatic monitoring systems. As a result, users often fail to identify food conditions in time, leading to persistent food waste problems [2].

Computer vision has emerged as a promising technology to tackle this issue through automatic object recognition. Among various algorithms, *You Only Look Once (YOLO)* stands out as one of the most effective approaches for real-time object detection due to its speed and high accuracy [3]. Previous studies have applied YOLO to agricultural product monitoring and food quality inspection, but its specific implementation in household refrigerators remains limited [4].

This study proposes the design of a food monitoring system using the YOLOv5 algorithm. The system employs a camera to detect and categorize food items inside the refrigerator, processes the detection results, and displays them through a user-connected application. It is also equipped with real-time notifications to remind users about food conditions. Accordingly, the objectives of this research are: (1) to develop a computer vision–based system capable of accurately classifying food items, (2) to implement the system on affordable embedded hardware for real-time inference, and (3) to evaluate the system's accuracy and reliability through performance testing.

This research is expected to provide an innovative solution to reduce food spoilage, support sustainable consumption, and assist households in managing food storage more effectively.

METHOD

As in Figure 1, the system is designed using a Raspberry Pi 4 as the main processing unit, connected to a monitor, mouse, and keyboard for control purposes. A webcam is placed above the

https://doi.org/10.25077/9.2.51-58.2025

refrigerator to capture images of food items, which are then processed using the YOLOv5 algorithm. The system is equipped with a mode selection button, which allows users to choose between entry mode and exit mode.

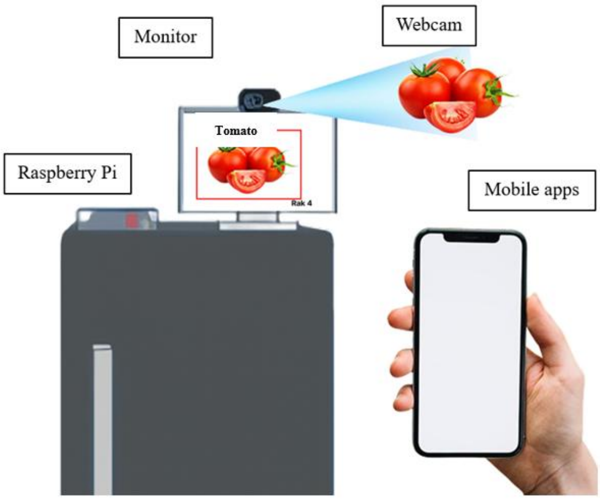


Figure 1 General System Design

In the entry mode, the Raspberry Pi processes input images captured by the webcam, detects the type of food, and displays the detection results on the monitor. The classification results are stored in a database and automatically sent to a Firebase-based application installed on the user's smartphone. The application then presents the food information and provides real-time notifications about food conditions.

In the exit mode, the Raspberry Pi once again processes input from the webcam to identify the food being removed from the refrigerator. The corresponding food data is deleted from the database and synchronized with the application, ensuring that the food inventory remains accurate. With these two modes, the system helps users manage food supplies more effectively and reduces the risk of food waste.

Software Design

The software design of the system is illustrated in the flowchart presented in Figure 2. The process begins with system installation and loading of the trained food dataset model using YOLOv5. Users then select the operation by pressing the mode button, choosing either entry mode or exit mode.

In entry mode, the webcam captures an image of the food, which is processed by the Raspberry Pi using YOLOv5. The detection results are displayed on the monitor, stored in the database, and transmitted to Firebase. The stored data is subsequently displayed in the mobile application and triggers a notification for the user.
In exit mode, the webcam captures an image of the food being removed. The system then deletes the corresponding food data from Firebase and the application, ensuring synchronization between the application and the actual contents of the refrigerator.

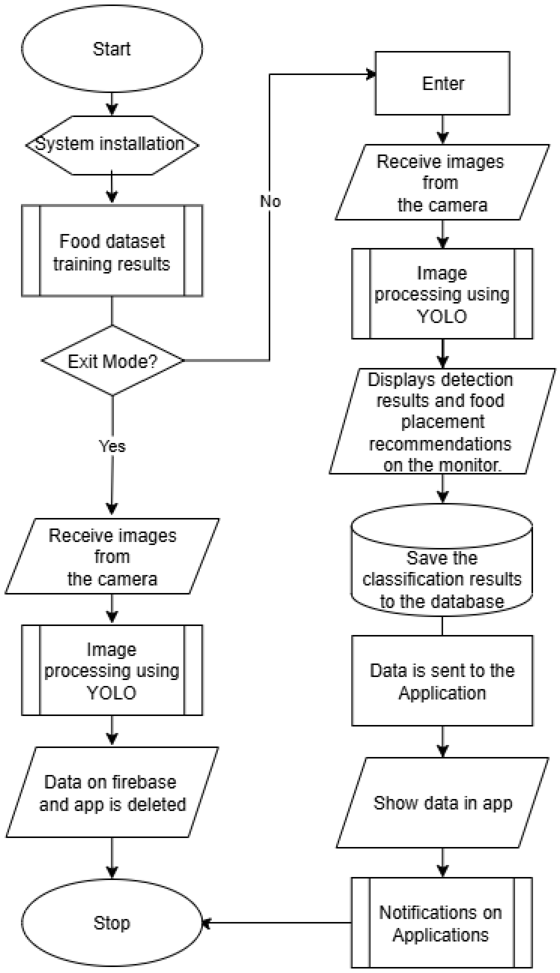


Figure 2 Software Design

Hardware Design

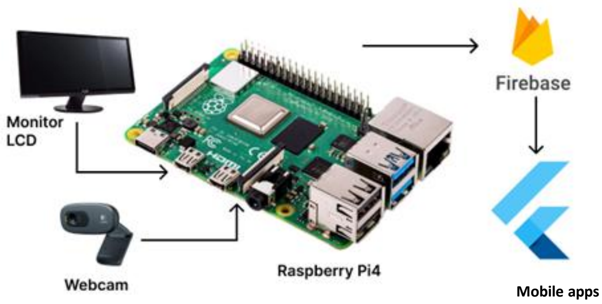


Figure 3 Hardware Design

The hardware components of the system consist of a Raspberry Pi 4, webcam, monitor, and a 3D-printed casing housing the main components. The casing is equipped with two mode buttons, distinguished by color: a green button for entry mode and an orange button for exit mode. Additionally, the casing includes a cooling fan to maintain the stability of the Raspberry Pi during computation.

As shown in Figure 3, the Raspberry Pi receives food image input from the webcam and processes it according to the programmed algorithm. The output consists of food detection results and notifications, which are sent to the user's application. This hardware design is intended to be user-friendly and practical for

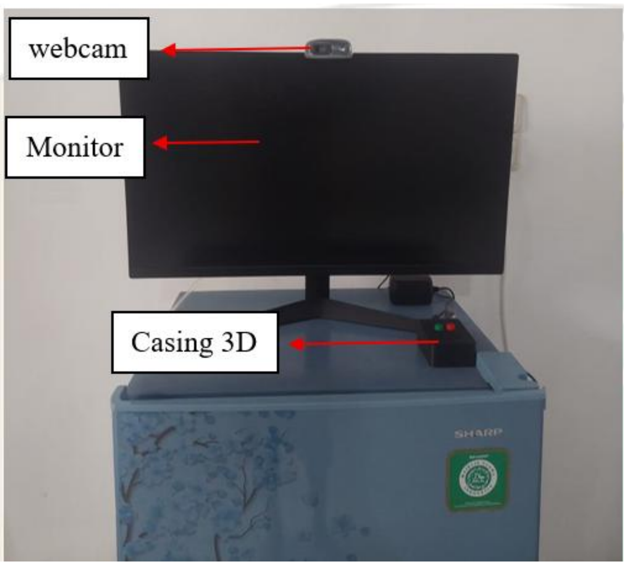
 https://doi.org/10.25077/9.2.51-58.2025

household use, combining a simple appearance with functional features.

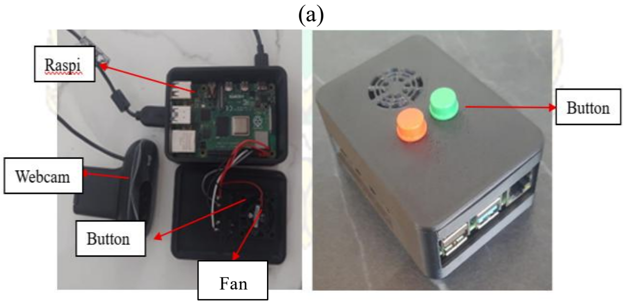
RESULTS AND DISCUSSION

Hardware Implementation

The system is built using several hardware components, including a Raspberry Pi 4, a Logitech webcam, an LCD monitor, a 3D printed casing with a cooling fan, and two mode buttons (green and red). The complete hardware configuration is shown in Figure 5.



(a)



(b)

Figure 4. Hardware :(a) Implementation (b) details

The green button functions as the entry mode, used when the user adds new food items into the refrigerator so that the system can detect, classify, and store the data in the database. Meanwhile, the red button serves as the exit mode, which is used to remove food data that is no longer available inside the refrigerator.

The Raspberry Pi 4 acts as the central controller of the system, running the YOLOv5 algorithm to detect food items from images captured by the webcam. The detection results are then displayed in real-time on the LCD monitor. To maintain temperature stability during intensive processing, the Raspberry Pi is placed in a 3D-printed casing equipped with a cooling fan. With this design, the system can operate independently, is easy to interact with using physical buttons, and ensures stable performance of the hardware during operation.

Software Implementation

In the software implementation, the system employs the YOLOv5s model as the main algorithm for food object detection. The model is executed using Python on the Raspberry Pi 4 Model B as the primary processing unit. Before deploying YOLOv5 on the Raspberry Pi, the food dataset was prepared and trained using Google Colaboratory with an NVIDIA Tesla T4 GPU.

The dataset consists of 12 food classes: Avocado, Spinach, Chili, Meat, Cabbage, Coconut-Milk-Based Dishes, Papaya, Milk, Eggplant, Tomato, and Carrot. Each class contains 200 images, resulting in a total of 2,400 images. The dataset is split into 70% training, 20% validation, and 10% testing sets. Augmentation techniques such as rotation, brightness adjustment, blur, cutout, and bounding box noise were applied, as in Figure 5.

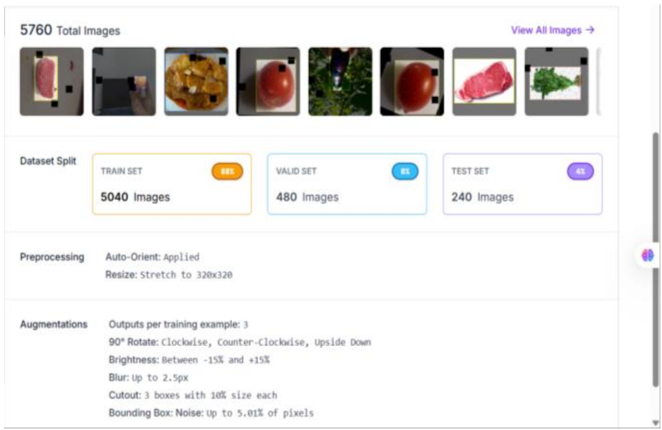


Figure 5. Dataset labeling process using Roboflow

The training was conducted for 50 epochs using the PyTorch framework. The model achieved strong performance, with *precision* > 0.90, *recall* > 0.90, and mAP@0.5 above 0.90, as shown in Figure 6. The best model was saved in .pt format and later converted to ONNX for lightweight execution on Raspberry Pi.

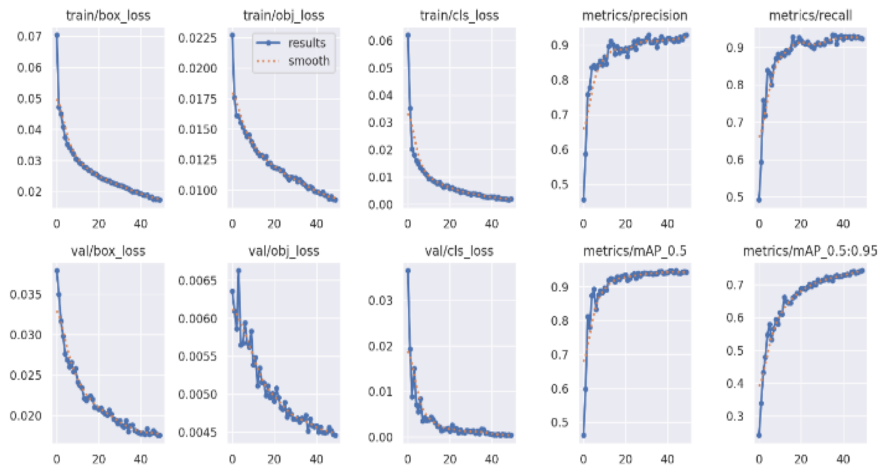


Figure 6. Training results of YOLOv5 model

Inference was carried out using *onnxruntime* with a Logitech C270 camera (720p). Each frame captured by the camera was normalized and passed into the model for detection. The output was displayed with bounding boxes and confidence scores.

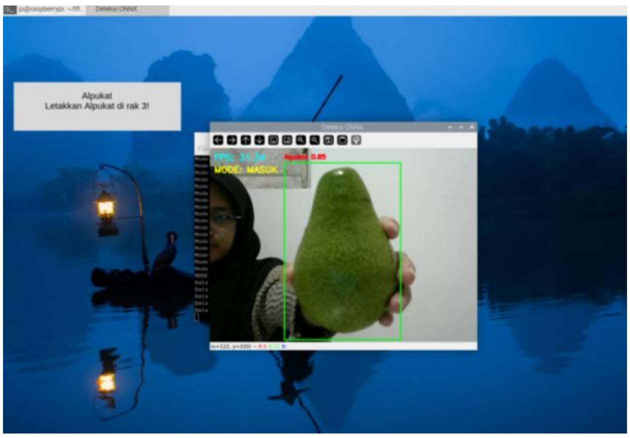
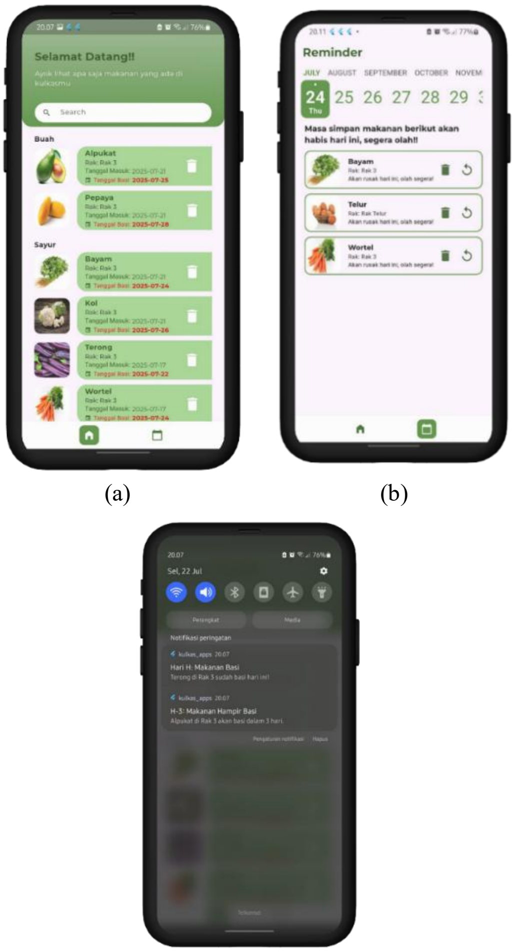


Figure 7. Example of food detection by YOLOv5

The system also provides local visual popups on the Raspberry Pi using Tkinter. These popups appear for 3 seconds to confirm successful detections in both input and output modes, displaying the detected object name and its recommended storage position. For example, in Figure 7, when an avocado is detected, the popup message reads: “Place the avocado on shelf 3” (“*Letakkan alpukat di rak 3*”). Detection results are stored in Firebase Firestore as JSON data and displayed on a Flutter-based Android application. The application has two main pages: (1) Food List, displaying the current food stock and expiration date, and (2) Reminder Calendar, showing items nearing expiration.

Additionally, the system generates push notifications on the Android status bar, which are automatically triggered at D–3, D–1, and on the expiration day (D-Day), as shown in Figure 8.



(a) (b)

(c)

Figure 8. Android application (a) Food list display on Android application (b) Reminder page (c) Notification display on Android application

 https://doi.org/10.25077/9.2.51-58.2025

TESTING AND ANALISYS

Testing Based on Object Distance and Position

The distance and position testing aimed to evaluate how well the system detects food objects at various distances and orientations relative to the camera. This evaluation is crucial to ensure reliability when implemented in real refrigerator environments.

The experiment was conducted using a Logitech C270 webcam connected to the Raspberry Pi 4. Distances were varied at 10 cm, 20 cm, 30 cm, 40 cm, and 50 cm, while object orientations were tested at frontal (front), left-side, and right-side positions.

Table 1. Testing Results of Object Distance and Position

Distance (cm)	Position	Average FPS	Detection Success
10	Front	45.47	58.3%
20	Front	32.81	91.7%
30	Front	36.81	100%
40	Front	33.40	100%
50	Front	35.11	66.7%
30–40	Left Side	33.14	100%
30–40	Right Side	34.27	100%
30–40	Front	37.01	100%

From the results, the system performed optimally at distances of 20–40 cm, with detection rates reaching 100% across frontal and side positions. At 10 cm, detection accuracy dropped due to limited camera focus, while at 50 cm the objects appeared too small for accurate recognition. Thus, the ideal operational distance is 20–40 cm, as in Table 1.

Testing Based on Light Intensity

The light intensity testing aimed to observe how variations in illumination affect object detection. Objects were placed at a fixed distance of 30 cm from the camera under two lighting conditions: 10 lux (low) and 45 lux (normal), measured using a Light Meter.

Table 2. Testing Results of Light Intensity

Light Intensity (lux)	Average FPS	Detection Success
10 (low)	25.45	50%
45 (normal)	34.57	100%

The results (Table 3) show that under 45 lux, the system achieved 100% detection accuracy with stable FPS above 30, ensuring real-time capability. At 10 lux, detection accuracy decreased to 50% and FPS dropped to 25.45, indicating that adequate illumination plays a critical role in detection performance.

Testing Object Detection Accuracy

The object detection accuracy test aimed to evaluate the system's reliability in correctly recognizing food items according to their respective classes. Testing was performed on 12 food classes with a total of 120 trials. Detection results were compared to ground truth labels to calculate True Positive (TP), False Positive (FP), and False Negative (FN).

Table 3. Testing Results of Detection Accuracy

No	Food Class	Test Trials	TP	FP	FN	Accuracy (%)
1	Avocado	10	9	0	1	90%
2	Spinach	10	10	0	0	100%
3	Chili	10	8	1	2	80%
4	Meat	10	9	0	1	90%
5	Cabbage	10	10	0	0	100%
6	Coconut Milk-based Food	10	10	0	0	100%
7	Papaya	10	10	0	0	100%
8	Milk	10	9	1	1	90%
9	Egg	10	10	0	0	100%
10	Eggplant	10	9	1	1	90%
11	Tomato	10	9	1	1	90%
12	Carrot	10	10	0	0	100%
	Total	120	113	4	7	94.17%

From Table 3, it can be observed that the system achieved an average detection accuracy of 94.17%. Several classes such as Spinach, Cabbage, Coconut Milk-based Food, Papaya, Egg, and Carrot were detected perfectly with an accuracy of 100%. However, certain classes such as Chili (80%) showed lower accuracy due to the relatively small object size and similarity with the background, which increased the number of False Negatives (FN).

Overall, this testing result demonstrates that the system performed reliably and achieved the targeted accuracy of above 90% as specified. Therefore, the proposed system can be considered dependable for detecting various food items stored inside the refrigerator and is ready for real-world household applications.

Overall System Testing

The overall system testing was conducted to evaluate the system's performance in real operational scenarios, specifically in two modes: Entry Mode and Exit Mode. This testing aimed to verify that the system not only detects food objects accurately but also integrates with Firebase and the Android application for proper data management, storage monitoring, and notification delivery.

Entry Mode

In Entry Mode, the user pressed the green button to activate the input process. The Logitech C270 camera captured the food image, which was then processed by the YOLOv5 model to detect the corresponding food class. If the detection remained stable for at least 2 seconds, the system displayed a confirmation popup containing the detected food name and the recommended storage shelf as in Figure 9.

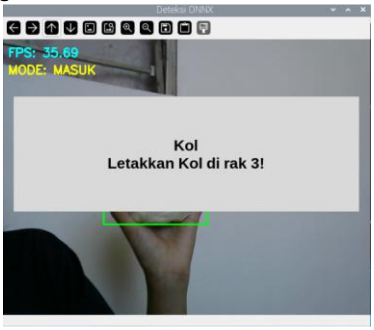


Figure 9. Entry mode detection process

Once confirmed, the following processes were performed:

1. The detection result was stored in Firebase, along with the calculated shelf-life expiration date (Figure 11).

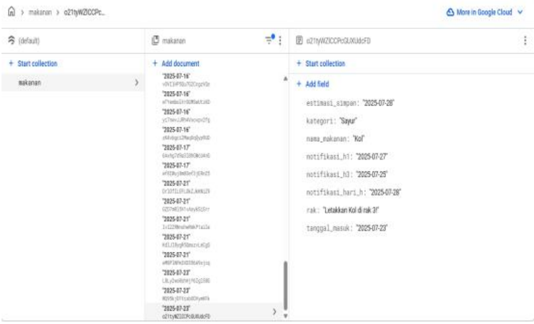


Figure 10. Data display in Firebase

2. The data stored in Firebase was forwarded to the Android application in real time.
3. The application displayed the detected food data together with its shelf-life estimation (Figure 11)

Figure 11. Data display in the application

4. Notifications were scheduled automatically based on expiration dates (D, D–1, D–3) (Figure 12).

Figure 12 Calendar display in the system

Exit Mode

In Exit Mode, the user pressed the red button to activate the removal process. The camera captured another food image, which was compared with entries in Firebase. If a match was found, the system executed the following:

1. The corresponding food entry was deleted automatically from Firebase, which also removed the record from the Android application (Figure 13).

Figure 13 Application display after detection with exit mode

2. A popup notification was displayed to confirm that the food item was successfully removed (Figure 14).

Figure 14 Detection display with exit mode

The overall testing demonstrated that the system worked in an integrated and consistent manner, following the designed

 https://doi.org/10.25077/9.2.51-58.2025

workflow. Both Entry and Exit Mode produced correct outputs without any errors or missed responses. The system was able to detect food items accurately, store and delete data properly, and provide reliable information to the user through the Android application. These results prove that the system fulfilled several elements in the House of Quality (HOQ), namely Accurate Data Processing, Real-time Processing, and Computational Method. In addition, the test verified the fulfillment of the main requirements, which include detecting food types, recording and displaying food data, maintaining real-time performance, and automatically deleting data when the food is removed. Thus, it can be concluded that the proposed system has successfully achieved its design objectives and is ready for practical implementation in daily household use.

CONCLUSIONS

In conclusion, the proposed YOLOv5-based food monitoring system demonstrated reliable and consistent performance in detecting and classifying food items, achieving an average accuracy of 94.17%, exceeding the predefined target of 90%. Experimental evaluation showed that the system performs optimally at object distances between 20–40 cm, where detection accuracy reached 100%. Performance degradation was observed at very close (10 cm) and farther (50 cm) distances due to camera focus limitations and reduced object scale. Lighting conditions were also found to significantly influence detection results; under normal illumination (45 lux), the system maintained 100% accuracy with stable real-time performance (FPS > 30), whereas low-light conditions (10 lux) led to a notable decline in both accuracy and frame rate.

Furthermore, the system was successfully implemented on a Raspberry Pi 4 and fully integrated with Firebase and an Android application to support real-time data synchronization and user notifications. The Entry Mode effectively detected, recorded, and displayed food items along with their estimated shelf-life, while the Exit Mode ensured automatic deletion of corresponding records when items were removed. These findings confirm the feasibility of deploying an embedded computer vision–based solution for intelligent refrigerator monitoring, contributing to practical smart home applications and supporting efforts to reduce household food waste.

REFERENCES

[1] M. A. P. Handoyo and N. P. Asri, “Study on Food Loss and Food Waste: Conditions, Impact and Solutions,” *AGRITEPA J. Ilmu dan Teknol. Pertan.*, vol. 10, no. 2, pp. 247–258, 2023, doi: 10.37676/agritepa.v10i2.4579.

[2] M. E. Mua, “Berapa Lama Sebaiknya Menyimpan Makanan Di Dalam Kulkas,” rri.co.id. Accessed: Feb. 04, 2025. [Online]. Available: https://www.rri.co.id/lain-lain/531243/berapa-lama-sebaiknya-menyimpan-makanan-di-dalam-kulkas

[3] F. Alfian, “Penyebab besarnya Angka Food Loos sampai Food Waste di Indonesia,” surabayaonline.co. Accessed: Feb. 02, 2025. [Online]. Available: https://surabayaonline.co/2021/10/15/penyebab-besarnya-angka-food-loos-sampai-food-waste-di-indonesia/

[4] A. Ramadhan, “Penelitian ungkap penyebab ‘Food waste’ di Indonesia tinggi,” antaranews.com. Accessed: Feb. 02, 2025. [Online]. Available: https://www.antaranews.com/berita/2453397/penelitian-ungkap-penyebab-food-waste-di-indonesia-tinggi

[5] A. Wahyuningsih, “Asal taruh bisa berisiko, cara tepat meletakkan bahan makanan di kulkas sesuai zona dan daya tahan,” briliofood. [Online]. Available: https://www.briliofood.net/foodpedia/asal-taruh-bisa-berisiko-cara-tepat-meletakkan-bahan-makanan-di-kulkas-sesuai-zona-dan-daya-tahan-241231s.html

[6] F. N. Kuaranita, “Ini Cara Mengatur Letak Makanan di Kulkas yang Tepat,” Kompas Klasika. [Online]. Available: https://klasika.kompas.id/baca/cara-mengatur-letak-makanan-di-kulkas/

[7] C. Budi, “Purwarupa Sistem Monitoring Makanan Kulkas Pintar Berbasis Internet Of Things,” 2021. [Online]. Available: https://elibrary.unikom.ac.id/id/eprint/5606/

[8] A. R. Adwiyah, B. Rahmat, and A. I. Irawan, “Kulkas Pintar Untuk Mendeteksi Kualitas Dan Kuantitas Bahan Makanan Melalui Media Android Menggunakan Mikrokontroler,” vol. 8, no. 5, pp. 5135–5142, 2021.

[9] A. W. Sutjiadi *et al.*, “YOLO”.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once : Unified , Real-Time Object Detection”.

[11] M. S. Hidayatulloh, “Sistem Pengenalan Wajah Menggunakan Metode Yolo (You Only Look Once),” *Univ. Med.*, pp. i–43, 2021.

[12] Y. Li, Y. Fan, S. Wang, J. Bai, and K. Li, “Application of YOLOv5 Based on Attention Mechanism and Receptive Field in Identifying Defects of Thangka Images,” *IEEE Access*, vol. 10, no. July, pp. 81597–81611, 2022, doi: 10.1109/ACCESS.2022.3195176.

[13] A. Sukusvieri, “Implementasi Metode Single Shot Detector untuk Pengenalan Wajah,” *Univ. Din.*, 2020.

[14] N. Krisna, “Memahami dan Menerapkan R-CNN yang Lebih Cepat: Panduan Langkah demi Langkah.” [Online]. Available: https://towardsdatascience.com/understanding-and-implementing-faster-r-cnn-a-step-by-step-guide-11acfff216b0

[15] K. C. Gross, C. Y. Wang, and M. E. (eds) Saltveit, “The commercial storage of Fruits, Vegetables, and Florist and Nursery Stocks,” *Agric. Res. Serv. United States Dep. Agric. Wahingt. DC*, vol. 66, no. 66, p. 442, 2016, [Online]. Available: www.ntis.gov.

[16] N. Asiah, L. Cempaka, and W. David, *Pendugaan Umur Simpan Produk Pangan Nurul Asiah , Laras Cempaka , Wahyudi David*, no. February. 2018.

AUTHOR(S) BIOGRAPHY

Laellatul Husna
Born in Padang Panjang, West Sumatera, on 1 March 2003, Then studied at the Department of Computer Engineering, Faculty of Information Technology, Andalas University