International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

Artificial Intelligence- and Internet of Things-Enabled Predictive Monitoring Systems: A Comparative Review with Application to FoodGuard AI

Adesh Tikande[1], Shreyash Mali[2], Harshal Borse[3], Dhiren Kamble[4], Samreen Shaikh[5], Samina Shaikh[6]

[1,2,3,4]*Department of Computer Engineering, Trinity College of Engineering and Research, Pune, India*
[5,6]*Assistant Professor, Trinity College of Engineering and Research, Pune, India*

***Abstract*—Every day, warehouses and quick-commerce dark stores lose a large amount of food to spoilage, and most of them still catch it by hand, after the damage is done. In this paper we review three recent predictive monitoring systems that use artificial intelligence and the Internet of Things, each from a different field: a fall-risk predictor for elderly people that pairs motion and vital-sign sensing with a gated recurrent unit running at the edge, a fruit freshness classifier that combines a convolutional neural network with environmental sensors, and an irrigation controller that compares four machine-learning models. The three systems have little in common on the surface, yet they share one design pattern: several sensors feed a learned model, and the result is turned into a real-time alert, often at the network edge so that cloud latency does not get in the way. Using this pattern, and three gaps we found in the reviewed work (no reading of expiry labels, no joint reasoning over visual and environmental signals, and no automated dispatch advice), we propose FoodGuard AI, a predictive food-safety and inventory-intelligence platform. It brings together optical character recognition of expiry dates, a YOLOv8 model for judging produce freshness, temperature and humidity readings from a microcontroller, and a shelf-life engine that produces first-expired-first-out dispatch recommendations on a live dashboard. We describe the architecture, the module-wise methodology, an eight-month implementation roadmap, a comparison of the reviewed literature, and the evaluation metrics we plan to use.**

***Keywords*—computer vision, deep learning, food waste reduction, Internet of Things, predictive analytics**

I. Introduction

A great deal of food is lost after harvest, while it sits in dark stores, supermarkets, and cold-storage units or travels the last mile to the customer. Produce is often thrown away only once spoilage has visibly spread. A temperature excursion inside a cold bin can go unnoticed until the stock is already damaged. Printed expiry dates are still read by eye during busy intake periods, which is slow and easy to get wrong [1]. The same reactive habit shows up in many other safety-critical settings, where a system responds only after something has gone wrong instead of warning ahead of time [2].

Advances in the Internet of Things (IoT) and Artificial Intelligence (AI) are changing this. Researchers have already moved from reacting to anticipating in areas as different as elderly fall-risk prediction [2], fruit and vegetable freshness grading [1], and irrigation control in precision agriculture [3]. The recipe is similar each time: inexpensive sensors keep collecting physical or physiological signals, and a machine-learning or deep-learning model turns those raw signals into an early warning, well before the fall, the spoilage, or the water stress can no longer be undone.

In this paper we look closely at three representative systems from those fields. Each one uses a different mix of sensors and AI to predict problems in real time. Our aim is twofold: (i) to pull out the architectural and algorithmic patterns that make such predictive IoT–AI systems work, and (ii) to use those patterns to shape FoodGuard AI, a predictive food-safety and inventory-intelligence platform that we are building as our final-year engineering project. The rest of the paper is organized as follows. Section II reviews the three systems along with related work on freshness classification, expiry-label recognition, and cold-chain management. Section III compares them. Section IV lists the research gaps that follow. Section V presents the FoodGuard AI architecture and methodology, Section VI gives the implementation roadmap, Section VII explains how we plan to evaluate the system, and Section VIII concludes.

II. Literature Survey

A note on the numbers in this section. Every accuracy, precision, recall, and sensor threshold quoted below was reported by the authors of the cited work. We have summarized and paraphrased them only to make comparison possible. We have not re-run or independently verified them, and we do not reproduce any figures, tables, or datasets from the original papers.

A. Multimodal Sensor Fusion for Proactive Fall-Risk Prediction

Harish et al. [2] built Elderly FallGuard, a wearable system that tries to predict that an elderly person is about to fall instead of simply detecting the fall afterwards.

An MPU6050 inertial measurement unit (IMU), which combines an accelerometer and a gyroscope, sits on the foot and records gait and movement. A MAX30100 sensor on the wrist tracks heart rate and blood-oxygen saturation (SpO2). The authors do not process the two data streams together from the start. They train and run two separate Gated Recurrent Unit (GRU) models, one for the IMU and one for the vitals, because gait is a discrete-motion signal while vital signs form a continuous physiological time series. A Multimodal Fall Prediction Algorithm (MFPA) then combines the two outputs. A finite-state machine (FSM) with a 30-second timer decides when to raise the alert, which makes a smart walking stick vibrate and sends a 4G or 5G notification to caregivers. All inference happens at the edge, so the device avoids the delay and battery drain of streaming data to the cloud. The vitals-based GRU reached 95.2% accuracy (96.5% F1-score) and the IMU-based GRU reached 99% accuracy, and both beat K-Nearest Neighbors and simpler recurrent baselines tested in the same study.

B. Deep Learning and Internet of Things-Based Food Quality Detection

Krishna et al. [1] surveyed machine-learning and IoT methods for judging food quality and then built a Convolutional Neural Network (CNN) that separates fresh from rotten apples, oranges, and bananas using a public fruit-image dataset. The network has two convolution-and-max-pooling blocks followed by a flatten layer and dense layers. After 15 epochs of training it reached 98.46% training accuracy and 96.59% validation accuracy. On the IoT side, the paper describes a low-cost cold-storage rig made of a Raspberry Pi, a DHT11 temperature-and-humidity sensor, and an MQ5 gas sensor that picks up the volatile compounds given off by spoiling food. An item counts as fresh only if the temperature is between 0 and 35 °C, the humidity is between 45% and 70%, and no spoilage gas is detected. The important point for our work is that the CNN and the sensor rig are built and evaluated as two separate subsystems. They are never merged into one decision, so what the camera sees and what the environment is doing at that moment are never considered together.

C. Internet of Things and Machine Learning for Smart Irrigation

Kadiyala et al. [3] describe a cost-effective IoT irrigation system for rice. It uses temperature, pressure, soil-moisture, and water-level sensors read through an Arduino, processed on a Raspberry Pi, and pushed to a cloud dashboard where a farmer can monitor the field and override the pump by hand.

The main contribution is a head-to-head test of four classifiers: an Artificial Neural Network (ANN), a Support Vector Machine (SVM), a Decision Tree (DT), and Naive Bayes (NB). They were trained on 3,200 labeled sensor readings to predict when the pump should switch on, with ground truth defined by thresholds (temperature above 30 °C, water level below 1 cm, moisture below 20%). The ANN scored highest, with 98.33% accuracy and a 98.3% F1-score. The Decision Tree followed with a 95.5% F1-score, and the authors point out that it is easier for non-technical users to interpret even though it is less accurate. Unlike [2], this system depends on the cloud for its dashboard and alerts instead of running inference at the edge, and unlike both [2] and [1], it has no computer-vision component.

D. Computer-Vision-Based Freshness Classification and Expiry-Label Recognition

Besides the CNN classifier in Section II-B, several researchers have applied You Only Look Once (YOLO) detectors directly to freshness grading. Mukhiddinov et al. [4] improved a YOLOv4 backbone with a Mish activation function so that it could locate and classify fruits and vegetables as fresh or rotten in one pass, and they report higher average precision than the original YOLOv3 and YOLOv4. Karakaya et al. [5] compared several convolutional architectures on the binary fresh-versus-rotten task, which helped make fruit freshness a standard benchmark problem. Oliullah et al. [6] went a step further by combining an EfficientNetB7–ResNet50 hybrid with IoT sensing and a web interface, so that freshness grading became a deployable, sensor-integrated system instead of an offline classifier. This is the direction we follow in FoodGuard AI (Section V). For reading labels, Florea and Rebedea [7] developed a deep-learning optical character recognition (OCR) pipeline tuned for expiry dates on food packaging. They showed that general-purpose OCR engines such as Tesseract [8] do poorly on this task without domain-specific training. Gong et al. [9] also used a deep neural network, first to find the expiry-date region on a package and then to read the characters, which made the method more robust against cluttered backgrounds. Together these studies show that expiry-date OCR is a separate and non-trivial problem from produce-freshness classification. That is why FoodGuard AI keeps label reading (Module 1) and visual freshness classification (Module 2) as two distinct modules whose outputs can be fused, instead of forcing both into one end-to-end model.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

E. Internet of Things-Enabled Cold-Chain and Warehouse Management Systems

On the logistics side, Alshdadi et al. [10] implemented a low-cost framework of sensors and microcontrollers that tracks temperature and humidity in real time across a whole cold supply chain, from warehouse storage to transport, with cloud visibility for every stakeholder. This covers far more ground than the single-bin monitoring discussed in Section II-B. Lee et al. [11] designed an IoT-based warehouse management system for smart logistics and showed that sensor-driven automation can noticeably improve picking and dispatch efficiency, which is what motivates the First-Expired-First-Out (FEFO) dispatch module in FoodGuard AI. Humayun et al. [12] surveyed IoT combined with blockchain for logistics and transportation and stressed traceability, which complements the real-time spoilage prediction we are concerned with.

At the packaging level, Katyal et al. [13] used machine vision to detect and correct defects in fluid-based packaging on a production line, showing that vision-based inspection is already common in industry well beyond produce grading. Finally, Ben-Daya et al. [14] reviewed how IoT is used in food-supply-chain quality management and concluded that most deployed systems do only one job, either monitoring the environment or tracing products, and do not combine predictive analytics with operational decision support. This is the integration gap identified in Section IV, and it is the one FoodGuard AI is meant to close.

III. Comparative Analysis Of Reviewed Systems

Table I sets the three reviewed systems side by side in terms of sensing, AI and machine-learning (ML) model, best reported accuracy, where processing happens, and main limitation. The three papers tackle unrelated problems, yet the same architecture appears in all of them: raw signals from two or more different sensors are collected, a learned model (not a fixed rule) turns them into a prediction, and a real-time notification brings a human decision-maker back into the loop. Fig. 1 shows how these patterns come together and how they inform the FoodGuard AI design described in Section V.

TABLE I
Comparative Summary of the Three Reviewed Systems

Feature	Elderly fall-risk prediction [2]	Fruit freshness detection [1]	Smart irrigation control [3]
Sensors and data	Foot IMU (accelerometer and gyroscope); wrist MAX30100 (heart rate, SpO2)	Camera images; DHT11 (temperature, humidity); MQ5 (gas)	Temperature, pressure, soil-moisture, and water-level sensors
AI and ML model	Dual-stream GRU with FSM (MFPA)	CNN (two convolution and pooling blocks)	ANN, SVM, DT, and NB (compared)
Best accuracy	95.2% (vitals); 99% (IMU)	96.59% (validation)	98.33% (ANN)
Processing	Edge	Local (Raspberry Pi)	Cloud (via Raspberry Pi)
Key limitation	Wearable compliance; IMU noise sensitivity	Vision and IoT sensing not fused; only three fruit types	Static thresholds; single crop; no vision

Table I also shows where each system does its processing. Edge inference is used where a fast response matters for safety, as in fall prediction [2]. Cloud processing is kept where decisions come more slowly and remote, multi-site monitoring is useful, as in irrigation scheduling [3]. The food-quality system [1] sits between the two: it processes data locally on a Raspberry Pi but has no real-time alerting loop like the one in [2]. Fig. 2 plots the best accuracy reported for the main model in each system. Please read it as a summary of what each paper reported for itself and not as a fair head-to-head benchmark, because the datasets, task difficulty, and class balance differ from paper to paper.

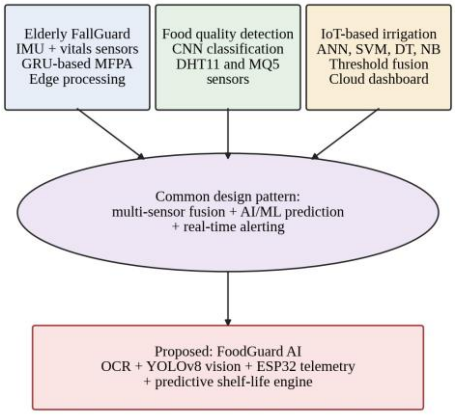


Fig. 1. Convergence of the design patterns identified across the three reviewed systems (multi-sensor fusion, AI and ML prediction, and real-time alerting) into the proposed FoodGuard AI architecture.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

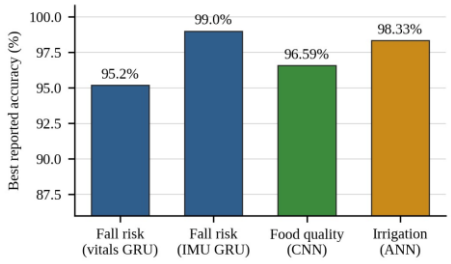


Fig. 2. Best reported accuracy of the primary AI and ML model in each reviewed system.

IV. Research Gaps And Motivation

Reading the three systems with a warehouse in mind, we noticed three gaps that none of them fully closes:

- *No expiry-label reading:* The CNN freshness classifier in [1] judges only how the produce looks. It cannot read a printed expiry date, so a packaged item that looks fine but has an expired label goes unflagged.
- *No joint visual-and-environmental reasoning:* In [1], the vision model and the environmental sensors (temperature, humidity, gas) are evaluated as separate subsystems instead of feeding one spoilage decision. The dual-stream fusion shown for physiological and motion data in [2] has no counterpart here.
- *No automated dispatch decision-support:* The systems in [1] and [3] produce a classification or a threshold-based control signal, but neither tells anyone what to do next, for example which stock to send out first. A warehouse manager needs exactly that kind of recommendation to act on a prediction.

These gaps shaped FoodGuard AI, which we describe next. It brings OCR-based label reading, a vision-based freshness classifier, IoT environmental telemetry, and a predictive shelf-life engine into one fused decision pipeline that also outputs a FEFO dispatch recommendation.

V. Proposed System: FoodGuard AI

A. System Overview

FoodGuard AI is a predictive food-safety and automated-warehouse platform aimed at quick-commerce dark stores, supermarket fresh-stock rotation, and cold-storage facilities. It has four sensing and inference modules: an OCR label reader, a produce-freshness classifier based on You Only Look Once version 8 (YOLOv8) [15], environmental telemetry from an ESP32 microcontroller, and a predictive shelf-life engine. A fusion-and-decision layer sits on top of them and issues FEFO dispatch recommendations and real-time alerts through a web dashboard. Together these address the three gaps listed in Section IV.

B. Hardware and Software Requirements

On the hardware side, an ESP32 microcontroller with built-in Wi-Fi sends telemetry from DHT22 or SHT31 temperature-and-humidity sensors placed inside the storage bins. A high-resolution camera, connected over universal serial bus (USB) or camera serial interface (CSI), captures images for both label OCR and produce-freshness checks. The vision pipeline uses OpenCV with a YOLOv8 model for freshness classification and EasyOCR or Tesseract to extract expiry dates. The backend is written in Python 3.10 or later and Node.js with Express. MongoDB stores real-time telemetry and stock logs, and a React.js dashboard handles visualization. The whole stack is meant to run on Windows 11 or Ubuntu Linux hosts.

C. System Architecture

Fig. 3 shows the layered architecture we propose. The Sensing Layer collects raw images and environmental readings. The Edge AI Layer then works on three tasks independently: it extracts the expiry date (OCR), classifies visual freshness (YOLOv8), and estimates the remaining shelf life from temperature and humidity trends (predictive ML engine). This independent-then-fused structure follows the dual-stream design that worked for mixed sensor data in [2]. The Fusion and Decision Module merges the three outputs into a single FEFO dispatch recommendation, which is saved to MongoDB and made available through the application programming interface (API) and the dashboard. Time-critical spoilage risks are also sent straight to an immediate alert channel.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

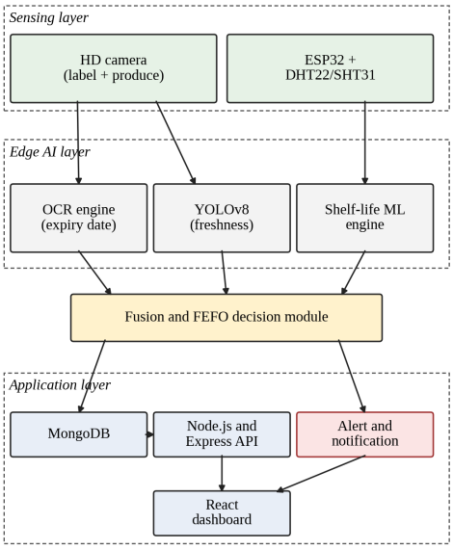


Fig. 3. Proposed layered system architecture of FoodGuard AI, showing the sensing, edge AI, fusion and decision, and application layers.

D. Module-Wise Methodology

The platform is divided into eight modules grouped into four development phases (Table II). Module 1 (OCR Label Scanner) reads and parses the expiry date printed on packaging. Module 2 (YOLOv8 Produce Freshness Classifier) grades incoming produce visually in real time. Module 3 (ESP32 Temperature and Humidity Sensing) continuously logs environmental readings for each bin. Module 4 (Predictive Shelf-Life Engine) takes the declared expiry date together with the observed temperature and humidity trend and recomputes how much shelf life each stock-keeping unit really has left, instead of trusting the printed date alone. Module 5 (FEFO Recommendation Engine) ranks stock by this adjusted shelf life to suggest what should be dispatched first. Module 6 (React Inventory Dashboard) shows warehouse staff the live stock status. Module 7 (Spatial Heat Map) displays temperature and humidity anomalies across the physical layout of the warehouse. Module 8 (Smart Notification System) sends alerts for items that are close to, or past, their predicted safe-use threshold.

VI. Implementation Roadmap

We have planned the work over eight months, phase by phase, as summarized in Table II. It starts with data ingestion and the AI inspection modules, moves on to IoT integration, predictive modeling, dispatch logic, and the dashboard, and ends with spatial visualization and alerting.

TABLE II
Proposed Eight-Month, Module-Wise Implementation Roadmap

Phase (months)	Modules	Deliverables
Months 1–2	Module 1: OCR Label Scanner Module 2: YOLOv8 Freshness Classifier	Ingestion pipeline and AI inspection modules operational
Months 3–4	Module 3: ESP32 Temperature and Humidity Setup Module 4: Predictive Shelf-Life Engine	IoT telemetry live; dynamic shelf-life recalculation model
Months 5–6	Module 5: FEFO Recommendation Engine Module 6: React Inventory Dashboard	AI dispatch logic and dashboard user interface integrated
Months 7–8	Module 7: Spatial Heat Map Module 8: Smart Notification System	Warehouse heat-map visualization and real-time alerts

VII. Expected Results And Evaluation Metrics

Comparable CNN- and GRU-based classifiers in the reviewed literature reach 96.59% for fruit-freshness classification in [1] and between 95.2% and 99% for physiological and motion prediction in [2]. We plan to evaluate the YOLOv8 freshness classifier and the predictive shelf-life engine with the same standard measures used in all three reviewed papers: accuracy, precision, recall, and F1-score, together with a confusion matrix. We want to be clear that FoodGuard AI is, at the time of writing, a proposed architecture and not a deployed, benchmarked system. The figures above are reference targets taken from earlier work and are not results we have obtained. Because the real goal of the platform is operational (cutting spoilage-driven write-offs) and not only predictive, we also propose two system-level metrics: (i) the mean absolute error between predicted and actual remaining shelf life for a sample of stock-keeping units, and (ii) end-to-end alert latency, measured from the moment a sensor threshold is crossed or a vision anomaly is detected until the notification appears on the dashboard. The second metric follows the edge-processing latency argument that [2] makes for safety-critical alerting.

VIII. Conclusion And Future Scope

In this paper we reviewed three recent AI- and IoT-based predictive monitoring systems from elderly care, food quality, and precision agriculture, and found a shared design pattern of multi-sensor fusion, AI- and ML-based prediction, and real-time alerting.

The reviewed food-quality and IoT-sensing work leaves three gaps open: expiry labels are not read, visual and environmental signals are not reasoned about together, and no automated dispatch advice is given. Building on the common pattern and these gaps, we proposed FoodGuard AI, a predictive food-safety and inventory-intelligence platform that fuses OCR, YOLOv8-based computer vision, ESP32 environmental telemetry, and a predictive shelf-life engine into a single FEFO-driven decision pipeline. Our next steps are to complete the eight-month roadmap in Table II and to benchmark the fused pipeline against the single-modality baselines reviewed in Section II. We also hope to add the center-of-pressure-style sensor-fusion refinements and the long-term real-world validation that [2] recommends as future work.

REFERENCES

[1] P. G. Krishna, P. S. M. Y, S. I, K. Swathi, M. K. B, and B. Suneetha, "A deep learning and IoT based food quality detection," in *Proc. 2023 Second Int. Conf. Electron. Renew. Syst. (ICEARS)*, 2023, pp. 577–583.

[2] H. S, N. Bhaskar, S. V, T. M, V. S, and V. Rajagopal, "Elderly FallGuard: An AI-driven assistive technology for fall risk prediction using motion and health monitoring," in *Proc. 2025 IEEE 22nd India Council Int. Conf. (INDICON)*, 2025.

[3] S. Kadiyala, A. Mehra, C. S. Potluri, S. Rajput, A. H. Birke, and K. Padmavathi, "An intelligent and cost-effective IoT-based irrigation system using machine learning," in *Proc. 2024 2nd Int. Conf. Disruptive Technol. (ICDT)*, 2024, pp. 380–385.

[4] M. Mukhiddinov, A. Muminov, and J. Cho, "Improved classification approach for fruits and vegetables freshness based on deep learning," *Sensors*, vol. 22, no. 21, Art. 8192, Oct. 2022, doi: 10.3390/s22218192.

[5] D. Karakaya, O. Ulucan, and M. Turkan, "A comparative analysis on fruit freshness classification," in *Proc. 2019 Innov. Intell. Syst. Appl. Conf. (ASYU)*, Izmir, Turkey, 2019, doi: 10.1109/ASYU48272.2019.8946385.

[6] K. Oliullah, M. R. Islam, J. I. Babar, M. A. N. Quraishi, M. M. Rahman, M. M. Rashid, and T. M. A. U. H. Bhuiyan, "FruVeg_MultiNet: A hybrid deep learning-enabled IoT system for fresh fruit and vegetable identification with web interface and customized blind glasses for visually impaired individuals," *J. Agric. Food Res.*, vol. 19, Art. 101623, 2025, doi: 10.1016/j.jafr.2024.101623.

[7] V. Florea and T. Rebedea, "Expiry date recognition using deep neural networks," *Int. J. User-Syst. Interact.*, vol. 13, no. 1, pp. 1–17, 2020, doi: 10.37789/ijusi.2020.13.1.1.

[8] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. Ninth Int. Conf. Document Anal. Recognit. (ICDAR)*, 2007, pp. 629–633.

[9] L. Gong, M. Yu, W. Duan, X. Ye, K. Gudmundsson, and M. Swainson, "A novel camera based approach for automatic expiry date detection and recognition on food packages," in *IFIP Adv. Inf. Commun. Technol.*, 2018, pp. 133–142, doi: 10.1007/978-3-319-92007-8_12.

[10] A. Alshdadi, S. Kamel, E. Alsolami, M. D. Lytras, and S. Boubaker, "An IoT smart system for cold supply chain storage and transportation management," *Eng., Technol. Appl. Sci. Res.*, vol. 14, no. 2, pp. 13167–13172, Apr. 2024, doi: 10.48084/etasr.6857.

[11] C. K. M. Lee, Y. Lv, K. K. H. Ng, W. Ho, and K. L. Choy, "Design and application of Internet of Things-based warehouse management system for smart logistics," *Int. J. Prod. Res.*, vol. 56, no. 8, pp. 2753–2768, Apr. 2018, doi: 10.1080/00207543.2017.1394592.

[12] M. Humayun, N. Jhanjhi, B. Hamid, and G. Ahmed, "Emerging smart logistics and transportation using IoT and blockchain," *IEEE Internet Things Mag.*, vol. 3, no. 2, pp. 58–62, Jun. 2020, doi: 10.1109/IOTM.0001.190009.

[13] N. Katyal, M. Kumar, P. Deshmukh, and N. Ruban, "Automated detection and rectification of defects in fluid-based packaging using machine vision," in *Proc. 2019 IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2019, pp. 1–5.

[14] M. Ben-Daya, E. Hassini, Z. Bahroun, and B. H. Banimfreg, "The role of Internet of Things in food supply chain quality management: A review," *Qual. Manag. J.*, vol. 28, no. 1, pp. 17–40, Dec. 2020.

[15] G. Jocher, A. Chaurasia, and J. Qiu, "YOLOv8," Ultralytics, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics. [Accessed: Sep. 10, 2026].