

IoT-Based Real-Time Banana Freshness and Spoilage Monitoring with Cloud Analytics and Push Notifications

Sabiya Banu H R
Department of ISE
AMC Engineering College
Bengaluru, India

Yashaswini C V
Department of ISE
AMC Engineering College
Bengaluru, India

Bathala Harshitha
Department of ISE
AMC Engineering College
Bengaluru, India

Divya Shree R
Department of ISE
AMC Engineering College
Bengaluru, India

Mohammed Ali Salam
Department of ISE
AMC Engineering College
Bengaluru, India

Abstract—Post harvest food loss in fresh food chains there is a need for strong early warning systems to allow preemptive action before large scale spoilage occurs. This indicates the need for affordable, granular monitoring tools. The proposed solution in this study addresses this need by using an ESP32 microcontroller, a DHT11 temperature and humidity sensor, and a CCS811 gas sensor to monitor ambient temperature, relative humidity, equivalent carbon dioxide (eCO_2), and total volatile organic compounds (TVOC) to implement and evaluate an Internet of Things (IoT) based solution. The fruit freshness is classified and rule-based shelf life estimates are provided by a multi parameter rule based algorithm implemented in a ReactJS dashboard. The dashboard links to a Node.js/Express backend that operates via WiFi. The dashboard leverages Google Firebase Firestore for information storage and utilizes Firebase Cloud Messaging to deliver alerts. The experimental findings indicate that the suggested multi parameter classification system is effective. For instance, a temperature of $27.5^{\circ}C$, relative humidity of 79%, eCO_2 of 452 ppm and TVOC of 7 ppb indicated an 80% rule based confidence indication that spoilage was imminent, estimating two days remaining. Conversely, a temperature of $26.4^{\circ}C$, a humidity level 72%, eCO_2 measurement of 589 ppm and TVOC value of 28 ppb suggested a "spoiled" condition. The monitoring measurements documented 3 instances, 123 spoiling occurrences, and 53 spoiled conditions. The system sent 28 alert messages to the user's device. These findings illustrate the viability of the affordable system for post harvest alerts, at the level of individual fruits.

Index Terms—IoT, monitoring fruit quality, detecting food spoilage, ESP32, CCS811 gas sensors, DHT11, Firebase, estimating shelf life, cloud technology, push alerts.

I. INTRODUCTION

Fruits and vegetables are perishable and a high percentage of what is harvested never reaches the consumer in an edible condition. According to the Food and Agriculture Organization, approximately one third of food produced for human consumption is lost or wasted annually, with postharvest losses of highly perishable produce being much greater, and disproportionately impacting regions with limited cold chain infrastructure [1] [2]. Bananas are fruits all over the world. Bananas continue to ripen after they have been picked. This process of maturation is affected by the respiration caused by ethylene, changes in ambient temperature and humidity, and the buildup of volatile organic compounds in the air [3] [4]. After the fruit exceeds the ideal ripening period, bacterial activity and enzymatic browning escalate swiftly, causing the fruit to spoil rapidly. Conventional quality evaluation depends on visual examination, which is subjective, labor intensive, inconsistent, and impractical for scaling across homes, store displays, or various storage locations [5] [6].

The Internet of Things (IoT) presents an alternative: affordable microcontrollers, environmental sensors, and cloud infrastructures can be combined to continuously and objectively monitor the micro climate surrounding stored goods, alerting to spoilage as it happens, rather than post event [7] [8]. Recent studies indicate that gas sensors detecting ethanol, methane, ammonia, and other volatile organic compounds effectively correlate with fruit ripening and decomposition stages. Integrating these signals with temperature and humidity data enhances the reliability of spoilage detection beyond using individual sensor types. This approach is much better because it

gives a complete picture of the situation [9] [4] [10]. However, despite this improvement, many current budgeting frameworks have limitations. They often do not have cloud storage support, trend analysis, and automatic user notifications. This hampers the process of monitoring the food quality and taking rapid action when problems arise. The following paper presents a system that addresses all these shortcomings and fills the gap in banana storage. Contributions of this paper include: This type of cost sensing device uses ESP32 chip and a sensor called DHT11 which can measure temperature and humidity, and a sensor called Adafruit CCS811 which can measure eCO₂ and TVOC values. This device sends information in JSON format to the server through WiFi connection in 5 seconds. There is a system with regulations on how to measure temperature, humidity, eCO₂, and TVOC values in order to get the thresholds. System determines whether the banana is fresh, overripe or rotten. This system predicts how many days the banana would last in the good condition and gives a score for this prediction. In addition to that, the Node.js/Express backend works with Firebase Cloud Messaging (FCM) in order to send push notifications on real time basis whenever there is any change in status or trajectory prediction of a fruit. The dashboard developed using React technology provides real time telemetry data. It provides statistics on the basis of the fleet and gives spoiling, spoiled, and warning counts. Perspectives for individual fruits are also provided along with a stream of current notifications. We evaluated our system with bananas.

II. LITERATURE SURVEY

A. Gas-Sensor-Based Spoilage Detection

Previous research and contemporary research show that cheap metal oxide gas sensors can be used for detecting spoiled fruits. Saha and Ali [9] devised a system using a Raspberry Pi Pico, DHT11 sensor for temperature and humidity measurement, and an MQ4 sensor for measuring methane to find out the level of spoilage of fruit and estimate the period of freshness of the fruit using regression mathematical technique on the data gathered by sensors. Other efforts using MQ sensors along with Arduino or NodeMCU microcontroller systems have been made for detecting ethylene, carbon monoxide, and ethanol gas coming from apples, mangoes, and bananas. These efforts send SMS or notifications through an application on reaching a certain threshold value [11] [12]. In some cases, cheap metal oxide gas sensors can still be used as a technique for monitoring spoilage of fruit. Bhattacharjee et al. A power nose that used a set of MQ gas sensors was developed by [4]. The tool was used to detect ethanol, methane, and ammonia emissions from bananas as they ripened and started rotting. Based on the gas concentration data, they developed a quality index using concentration thresholds. Tyagi et al. [10] created an affordable electronic nose for assessing fruit maturity. Fruit freshness devices also use MQ3 alcohol sensor in combination with cloud dashboard tools like Blynk. The devices have been used to track fruits and vegetables [13] [14].

B. Combining Sensor Data for Predictive Analysis

Recent researches employ signals and machine learning in order to improve banana ripeness predictions. Rajini and Voola [15] proposed a machine learning model to predict the ripeness and decay of bananas. For their experiment Rajini and Voola used sensors of temperature, humidity and gas emission connected to ESP32 microcontroller. Rajini and Voola also tried different machine learning classifiers in their research. The results of Rajini and Voola showed that among all the classifiers that they examined, CatBoost had the highest accuracy rate. Saranya and Srinivasan [3] used machine learning to estimate ripeness levels of bananas, while Chuquimarca et al. [16] tested the robustness of banana ripeness classifiers in different lighting conditions, highlighting the sensitivity of image processing approaches to environmental parameters and hence encouraging further development of other non visual approaches based on gas detection and environmental sensing techniques. Studies using vision based approaches, such as the application of transformer models and multi task networks for fresh food classification and literature review on automated ripeness classification of fruits [5] also reveal the progress made in machine learning and sensor fusion approaches. This survey deals with imaging computer vision and sensors based approaches that are non destructive in nature. In addition to this, practical approaches to fruit quality assessment have also proposed the integration of sensing technology with machine learning.

C. Cloud Connectivity and Notification Infrastructure

Continuous sensor data transmission to the cloud and immediate alert notifications to users are essential in configurations. Firebase Realtime Database and Firestore are frequently selected as backends for IoT data because they facilitate real time updates among devices without requiring an additional server [18]. Hasib and Akib [19] showcased an ESP32 setup for over seeing and controlling devices remotely. This Firebase system demonstrated that data transfer was exceptionally dependable and that control commands arrived at their destination in less than two seconds. The design employed a DHT sensor along with cloud services, which is quite similar to the configuration utilized here. Due to these outcomes, we selected Firestore and Firebase Cloud Messaging to ensure data security and to deliver alerts promptly.

D. Smart Agriculture and Cold Chain Monitoring

Internet of Things (IoT) and Wireless Sensor Networks have been widely studied and employed as tools for intelligent agriculture [7][8]. IoT and Wireless Sensor Networks have been employed in tasks carried out before, during and after harvesting [20]. An analysis of the Cold Chain and the Postharvest process suggests that precise measurement of temperature, humidity, and gases is important. It is necessary to reduce loss of fruits and vegetables during the period of storage, transportation, and retail [2][21][24]. However, household spoilage monitoring has not been given due importance according to these analyses. Large scale cold chain logistics operations are

extensively emphasized. The difference between this particular cold chain logistics approach and other approaches is that the present cold chain logistics process uses four unique signals generated by two affordable digital sensors on one ESP32 chip rather than just one analog gas sensor. The rule based scoring system used for this cold chain logistics process is very easy to understand and works well within a household with little or no labeled training data available. All the data is constantly stored in a NoSQL database with all the details of each parameter.

III. SYSTEM ARCHITECTURE

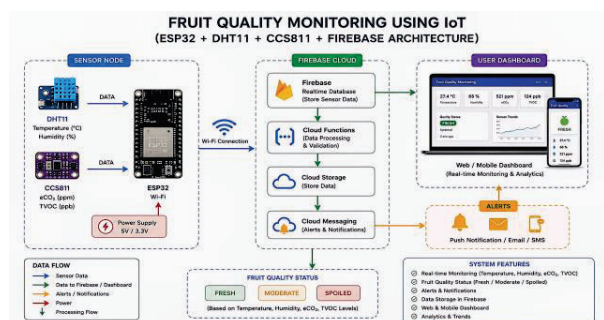


Fig. 1. System architecture diagram showing the end to end data flow across the sensing node (ESP32, DHT11, CCS811), Firebase Cloud services, and user dashboards.

A. Sensing Tier

The key component of the sensing node is the ESP32 microcontroller board which was chosen because of the integrated WiFi capability and two core processing together with affordability. There is a DHT11 digital sensor, which is attached to the GPIO pin and provides information on the ambient temperature (in °C) and relative humidity (in %). Besides, there is an Adafruit CCS811 digital metal oxide gas sensor, which works via the I2C interface (SDA/GPIO21, SCL/GPIO22) providing equivalent CO₂ (eCO₂ from 400 to 8192 ppm) and total volatile organic compounds (TVOC from 0 to 1187 ppb). The provided parameters can be used as efficient and cheap indicators of the volatile emissions associated with fruit ripening and decomposition [4]. When the node boots, the firmware initializes the I2C bus, enables the sensors and WiFi connection to the chosen access point, including the automatic reconnect functionality. Every 5 seconds (the time can be set by the SEND_INTERVAL constant), the firmware receives all 4 parameters, prints them in the serial console and if both sensors return valid values, packs the results in JSON format and sends HTTP POST request to the backend's /api/sensor data endpoint, reestablishing WiFi connection when necessary.

B. Cloud Backend Tier

The back end is developed on Node.js and Express and provides a small REST API. On receiving a POST request containing sensor data, the back end performs a number of actions first, it validates that the data in all four fields is

present and that it is numeric second, it computes the current status, future status, days left, and confidence level using the scoring algorithm described in section IV; third, it checks the most recent data stored in Firestore for the same type of fruit to determine whether there has been any change in status; Afterwards, it stores a structured record, which contains raw values, the calculated state, the summary per each parameter, the forecasted state and the server time stamp in the sensorData collection; at last, if the service identifies a state change or first reading suggesting spoiled or spoiling fruits, it collects all available Firebase Cloud Messaging (FCM) tokens from the fcmTokens collection and notifies each one of the users through a named notification, removing all invalid tokens. Besides, there is another API endpoint, /api/notifications/register token, through which clients can register their FCM token.

C. Presentation Tier

The React based single page web application features three main views, namely the Overview view, where the count of fruits at the fleet level as fresh, spoiling, spoiled, and warning is depicted, along with the real time alerts feed (please see Fig. 1); the Fruit Details view, with sensors' data cards for each fruit, a circle representing "Days Remaining" and a detail panel with predictions, probabilities, and actions (please see Fig. 2), and lastly, the support views for Analytics, Profile, and Settings. In order to enable real time notifications to the user, the dashboard asks the browser for FCM permissions.

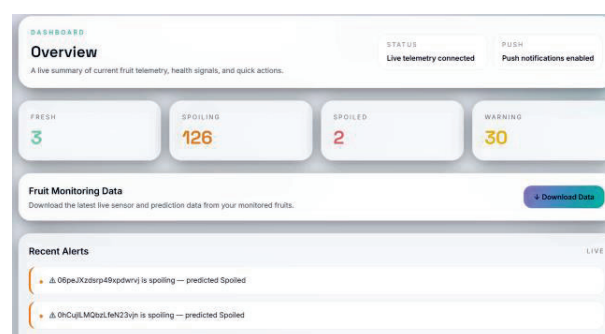


Fig. 2. Fruit Details view showing individual sensor cards (temperature, humidity, eCO₂, TVOC), predicted status, days remaining, and a detail panel with confidence and recommended action for the selected fruit.

IV. METHODOLOGY: SPOILAGE CLASSIFICATION AND SHELF LIFE ESTIMATION

A. Threshold-Based Parameter Scoring

With respect to the four parameters, namely, temperature, humidity, eCO₂ and TVOC, there is an elaborately crafted three band threshold table maintained by the backend, which is specific for bananas and is shown in Table I. For any reading x_p of a parameter p , the status s_p (fresh, spoiling, or spoiled) is determined based on the identification of the band that it belongs to. In case it does not belong to any of the bands, we take a conservative view and mark it as spoiled.

TABLE I
 BANANA THRESHOLD BANDS USED FOR PARAMETER SCORING

Parameter	Fresh	Spoiling	Spoiled
Temperature (°C)	13–18	18.1–25	25.1–40
Humidity (%)	85–95	75–84.9	0–74.9
eCO ₂ (ppm)	400–1000	1000.1–2000	2000.1–10000
TVOC (ppb)	0–150	150.1–500	500.1–5000

B. Composite Scoring and Overall Status

An integer penalty score S is accumulated across the four per-parameter statuses:

$$S = \sum_{p \in \{T, H, CO_2, TVOC\}} w(s_p), \quad w(s_p) = \begin{cases} 0 & s_p = \text{fresh} \\ 1 & s_p = \text{spoiling} \\ 2 & s_p = \text{spoiled} \end{cases} \quad (1)$$

The overall banana status is then determined from S as

$$\text{status} = \begin{cases} \text{fresh} & S = 0 \\ \text{spoiling} & 1 \leq S \leq 3 \\ \text{spoiled} & S \geq 4 \end{cases} \quad (2)$$

This method ensures that a combination of one highly deviating parameter (for instance, TVOC reading of a spoiled range with a weight of 2, along with one spoiling range parameter) will be sufficient to ensure that the fruit is no longer considered fresh, even though the rest of the parameters may be normal, although the fruit will only be considered completely spoiled when several highly deviated parameters exist together.

C. Shelf Life Estimation and Confidence

Given the overall status, the system reports a coarse estimated remaining shelf life and an associated confidence score, and separately predicts the fruit's next status:

- *fresh* : 5 estimated days remaining, predicted next status *spoiling*, 85% confidence;
- *spoiling* : 2 estimated days remaining, predicted next status *spoiled*, 80% confidence;
- *spoiled* : 0 estimated days remaining, predicted next status *spoiled*, 95% confidence.

Explanation about what these percentages stand for. The percentages 85%, 80%, and 95% are heuristic thresholds which are manually assigned and are constant for the whole status class that they belong to. These percentages do not come from any statistical or probabilistic analysis, nor have they been calibrated based on any error rate measure. These values are simply supposed to convey the idea that classification that is based on the fulfillment of several conditions at once (e.g., the spoiled class, which has, according to Eq. 2, $S \geq 4$) is more conclusive than one which is based on the crossing of a single threshold. In the same way, the estimated days remaining values are generated through the rules based score calculation algorithm described above and not through any prediction based on a model that is either trained or validated using observations of time to spoilage. No ground truth shelf life data in other words, the exact date when each banana

under observation was determined to have become spoiled was gathered during this research, meaning that the accuracy of these estimates has yet to be measured and thus should not be quoted as a validated value at this time.

D. Notification Logic

In order to avoid alert fatigue, a push notification is sent from the backend in the case of (i) where the new status is not the same as the old status for the same fruit; (ii) where the new next status differs from the old next status, or (iii) it is the first status for that particular fruit and it indicates that the fruit is spoiled or spoiling.

V. IMPLEMENTATION

Firmware is coded in C++ under the Arduino framework and compiled using PlatformIO for the ESP32 board. It uses the Adafruit CCS811 library for the gas sensor and a regular DHT library for the DHT11 sensor. In order to make it easy to deploy the code on various nodes, all important parameters such as WiFi credentials, pin definitions, backend URL, time between samples, and baud rate are gathered in one header file. At the backend side, we use Express 5 for routing, firebase admin SDK for access to Firestore and FCM, cors for access to dashboard, and dotenv for config. Every document under the sensorData collection includes the type of fruit, four values, the overall status, parameter status per individual parameter value, predicted status, confidence level, number of estimated days left, the flag indicating a notification has been sent, and the time when this was created on the server, thus allowing access to previous data. The dashboard is a React app that was created with the help of Vite and styled with the use of Tailwind CSS. It retrieves information from the backend/Firestore, computes and renders the count of fresh, spoiling, spoiled, and warning statuses and triggers alerts based on set thresholds.

VI. RESULTS AND DISCUSSION

This evaluation was conducted through the use of bananas for testing the performance of the sensing pipeline, classification, and end to end process from the firmware to the dashboard.

A. Representative Sensor Readings and Classification

Table II summarizes two representative readings captured directly from the deployed system: one from the on device serial console log and one from the corresponding Firestore document and dashboard detail view.

For Sample A, the temperature (27.5 °C) lies within the spoiled range and the humidity (79%) within the spoiling range while both the gas measurements (eCO₂ 452 ppm, TVOC 7 ppb) are still within the fresh range. Using Eq. (1), we obtain $S = 2 + 1 + 0 + 0 = 3$. From Eq. (2), the value of $S = 3$ lies between $1 \leq S \leq 3$, which corresponds to classification as spoiling, as in "Overall status: spoiling" reported in Firestore, shown in the Fruit Details page (Fig. 2), and summarized in Table II above. For Sample B, drawn from the serial log

TABLE II
 REPRESENTATIVE CAPTURED READINGS AND DERIVED CLASSIFICATION

Field	Sample A	Sample B
Temperature (°C)	27.5	26.4
Humidity (%)	79	72
eCO ₂ (ppm)	452	589
TVOC (ppb)	7	28
Temperature status	spoiled	spoiled
Humidity status	spoiling	spoiling
eCO ₂ status	fresh	spoiling
TVOC status	fresh	fresh
Overall status	spoiling	spoiled
Predicted status	spoiled	spoiled
Confidence	80%	95%
Est. days remaining	2	0

(Fig. 3, left), the eCO₂ value (589 ppm) has become within the spoiling range, along with a temperature measurement (26.4 °C). Now we have that $S = 2 + 1 + 1 + 0 = 4$, which surpasses the limit of $S \geq 4$ and is therefore categorized as spoiled just like the console log entry “Previous status: spoiled, Current status: spoiled, Predicted status: spoiled” suggests. Both examples show how the additive multi parameter scoring system identifies the difference between an early stage of spoilage (Sample A: overall spoiling, approximately 2 more days of shelf life) and an advanced stage of spoilage (Sample B: overall spoiled, 0 more days remaining). This is done using just affordable digital sensors and no trained classifier at all.

B. Data Persistence and Traceability

Fig. 3 (right) shows a representative Firestore document from the `sensorData` collection, confirming that every reading is persisted with its full per parameter breakdown, predicted status, confidence, and timestamp alongside the raw values. This structured storage enables retrospective auditing of which parameter(s) drove a given decision, useful both for tuning thresholds and for building a labeled dataset that could later support a supervised classifier, following the direction taken by [15].

C. Aggregate System Operation and Discussion

The dashboard Overview page (Fig. 1) indicates that within the period of monitoring, the system has logged 3 new readings, 123 readings that are spoiling and 53 readings that have already become spoiled. Of the above mentioned, 28 were marked with high severity warning because of some problems such as high humidity, TVOC and/or eCO₂ values. The large number of spoiling readings is quite in line with the expected development of the bananas from being ripe to becoming spoiled. The states logged also indicate that during the monitoring period there were readings that were spoiled. The recent alerts feed supports this statement and proves the effectiveness of our event driven alert system by sending notifications about fruit documents at the very moment of status change, not every 5 seconds.

This results in three main conclusions. The first one is that the combination of the temperature and humidity levels

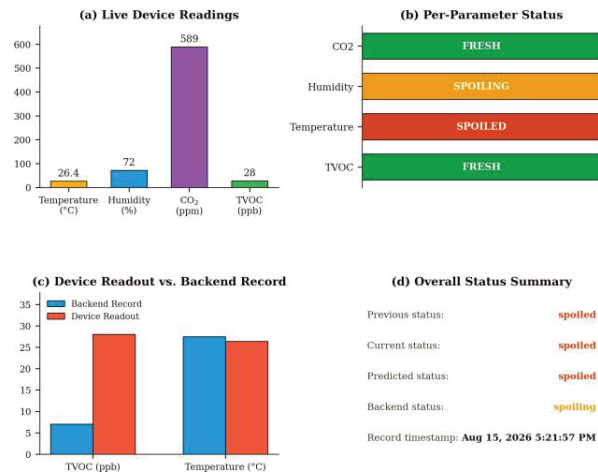


Fig. 3. Left: on device serial console log for Sample B, showing status transition tracking and raw readings. Right: the corresponding structured Firestore document, including per parameter status, predicted status, confidence, and timestamp.

along with eCO₂ and TVOC values allows detecting spoilage threats even when the value of any of the parameters alone is ambiguous. This confirms the concept of sensor fusion in the context of the e nose monitoring described in [4] and [10]. The second one is that the rule based approach proposed here, although simpler than classifiers described in [3] and [15], is fully transparent since each decision can be attributed to the corresponding parameter that exceeded the threshold level. Finally, the combination of cloud persistence and push notifications ensures a smooth transition from sensing to alerting that solves the problem left unaddressed by prior work which only had to deal with single sensors and dashboards [12], [13]. The thresholds set up in this paper have been optimized specifically for bananas and the sensor placements utilized in the experiment for bananas; modifying this process to cater to other fruits would involve setting up new threshold tables for the new fruit or switching to a learnable classifier using the structured Firestore database in a fashion analogous to [15].

VII. CONCLUSION AND FUTURE WORK

The following paper outlines an IoT enabled system created for banana quality and deterioration detection in real time. The system makes use of an ESP32 sensor module with integrated DHT11 and CCS811 sensors, a Node.js / Express backend with data persistence using Firebase Firestore, and push notifications using Firebase Cloud Messaging (FCM) via React based monitoring dashboard. A rule based multi parameter score algorithm is implemented to evaluate fruit state with the help of temperature, humidity, eCO₂, and TVOC measurements, providing rule based estimations of shelf life left. Banana samples were tested, and the working of the proposed multi parameter classification algorithm was demonstrated. The sensor readings as well as status information were always kept by the system on Firestore, whereas the

event driven mechanism would create an alert in case there was a change in status of relevance. In total during the monitoring process, the system recorded 3 new fruits, 123 spoiling fruits, and 53 spoiled fruits, producing 28 warnings. This shows how feasible the proposed low cost architecture can be for individual fruit monitoring, but more independent verification is needed to measure accuracy of classifications and shelf life estimations. Areas for further research include: (i) the extension of per fruit threshold configuration to other fruits, including apples, oranges, and mangoes, which are now implemented as placeholder types; (ii) the improvement of the rule based classifier using a supervised learner, such as CatBoost [15], trained using a growing annotated dataset gathered using Firestore; (iii) the inclusion of environmental sensors into ripeness detection using imagery following the methods presented in [5] and [16]; and (iv) the analysis of power consumption in order to facilitate battery powered deployments in a field and retail setting, unlike the current mains power setup.

REFERENCES

- [1] Food and Agriculture Organization of the United Nations, "Sustainable food cold chains: Opportunities, challenges and the way forward," FAO, Rome, Italy, 2021.
- [2] "Smart cold chain logistics for fresh agricultural products: key technologies, challenges, and future trends," *Trends in Food Science & Technology*, 2025, Elsevier ScienceDirect.
- [3] N. Saranya and K. Srinivasan, "Banana ripeness stage identification: a deep learning approach," *Journal of Ambient Intelligence and Human-ized Computing*, 2021, doi: 10.1007/s12652-021-03267-w.
- [4] A. Bhattacharjee, N. A. Biswas, K. A. Shahriar, and K. B. Salim, "IoT-based Cost-Effective Fruit Quality Monitoring System using Electronic Nose," arXiv:2512.08753, 2025.
- [5] "Insights into recent developments and obstacles in automated fruit ripeness classification," *e-Prime – Advances in Electrical Engineering, Electronics and Energy*, 2025, Elsevier ScienceDirect.
- [6] "A Mini Review on Sensor and Artificial Intelligence Approaches for Ripeness Detection and Classification of Oil Palm Fresh Fruit Bunch," *International Journal of Research and Innovation in Social Science*, 2025.
- [7] M. N. Mowla *et al.*, "Internet of Things and Wireless Sensor Networks for Smart Agriculture Applications: A Survey," *IEEE Access*, 2023, doi: 10.1109/ACCESS.2023.3346299.
- [8] "IoT-Enabled Smart Agriculture: Architecture, Applications, and Challenges," *Applied Sciences*, vol. 12, no. 7, p. 3396, 2022, MDPI.
- [9] A. Saha and L. Ali, "IoT Based Fruit Quality Inspection and Lifespan Detection System," in *Proc. 2023 7th Int. Conf. Computing, Communication, Control and Automation (ICCUBEA)*, 2023, doi: 10.1109/ICCUBEA58933.2023.10392254.
- [10] P. Tyagi, R. Semwal, and U. S. Tiwary, "E-nose: a low-cost fruit ripeness monitoring system," *Journal of Agricultural Engineering*, 2022, doi: 10.4081/jae.2022.1389.
- [11] "IoT Based Smart Fruit Spoilage Detection," *International Journal of Scientific Development and Research (IJS DR)*, vol. 10, no. 6, 2025.
- [12] "IoT Based Fruits Spoilage Detection using IoT," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 2023, doi: 10.22214/ijraset.2023.56101.
- [13] "Fruit Freshness Detection and Monitoring using IoT," *International Research Journal of Modernization in Engineering, Technology and Science (IRJMETS)*, 2024.
- [14] "Fruit Freshness Detection Using IoT," *International Research Journal of Modernization in Engineering, Technology and Science (IRJMETS)*, 2024.
- [15] M. Rajini and P. Voola, "Developing an IoT and ML-driven platform for fruit ripeness evaluation and spoilage detection: A case study on bananas," *e-Prime – Advances in Electrical Engineering, Electronics and Energy*, 2025, doi: 10.1016/j.prime.2025.100896.
- [16] L. E. Chuquimarca, B. X. Vintimilla, and S. A. Velastin, "Assessing deep learning model robustness for banana ripeness classification under varying illumination conditions," *e-Prime – Advances in Electrical Engineering, Electronics and Energy*, 2025, Elsevier ScienceDirect.
- [17] "Leveraging IoT and Machine Learning for Automated Fruit Quality Monitoring: A Scalable Approach for Supply Chain Optimization," *International Journal of Computer Applications*, vol. 187, no. 36, 2025.
- [18] "How To Integrate Firebase Real-time Database With Your IoT De-vice?," *Scientific World*, 2023.
- [19] A. Hasib and A. S. M. A. S. Akib, "Cloud-Enabled IoT System for Real-Time Environmental Monitoring and Remote Device Control Using Firebase," arXiv:2601.17414, 2026.
- [20] T. Miller, G. Mikiciuk, I. Durluk, M. Mikiciuk, A. Łobodzin'ska, and M. Śnieg, "The IoT and AI in Agriculture: The Time Is Now – A Systematic Review of Smart Sensing Technologies," *Sensors*, vol. 25, no. 12, p. 3583, 2025, doi: 10.3390/s25123583.
- [21] "Overview of Food Preservation and Traceability Technology in the Smart Cold Chain System," *Foods*, 2023, PMC.
- [22] "Sustainable Postharvest Innovations for Fruits and Vegetables: A Comprehensive Review," *Foods*, 2026, PMC.
- [23] "Food loss and waste management in the retail food supply chain: Methods and framework to achieve environmental sustainability," *Journal of Cleaner Production*, 2025, Elsevier ScienceDirect.
- [24] "The role of cold chain logistics in reducing postharvest losses," *Trends in Food Science & Technology*, 2026, Elsevier ScienceDirect.