© April 2026 | IJIRT | Volume 12 Issue 11 | ISSN: 2349-6002

IoT-AI Food Freshness Monitoring System

Renuka Ballal[1], Sakshi Khamankar[2], Sharayu Deote[3]
[1]*Department of Computer Engineering, Cummins College of Engineering for Women Nagpur, Hingna*
[2]*Assistant Professor, Computer Engineering Cummins College of Engineering for Women Nagpur, Hingna*
[3]*Head of Department, Computer Engineering Cummins College of Engineering for Women Nagpur, Hingna*

***Abstract*—This research proposes a Smart Food Monitoring System that integrates IoT sensors and AI-based image recognition to detect spoilage and ensure food quality in real-time. The system combines environmental sensors—such as gas, temperature, humidity, and motion—with a deep learning model based on Convolutional Neural Networks (CNNs) to assess food freshness visually. Data fusion from sensors and images provides accurate, early detection of spoilage, reducing food waste and enhancing consumer safety. A web application is developed to visualize data and generate alerts, offering a cost-effective and scalable solution for real-time food monitoring.**

***Index Terms*—Smart Food Monitoring, IoT Sensors, AI-Based Image Recognition, Food Spoilage Detection, CNN, Sensor Fusion, Food Safety.**

I. INTRODUCTION

Food safety has become a global concern due to increasing health risks and food waste. Traditional food inspection methods are manual, time-consuming, and ineffective for real-time monitoring. Emerging technologies like the Internet of Things (IoT) and Artificial Intelligence (AI) offer promising solutions by enabling automation, predictive analysis, and remote tracking of food quality.

This project integrates IoT-based environmental monitoring with AI-driven image recognition to detect food spoilage. Sensors such as DHT11 for temperature/humidity, MQ-series for gas emissions, and PIR sensors for energy optimization monitor storage conditions. Simultaneously, a CNN model analyzes visual cues like discoloration and mold to assess freshness. The system sends real-time updates and alerts to a user-friendly web application.

II. EASE OF USE

The Smart Food Monitoring System is designed with user-friendliness at its core. Users interact with a clean and intuitive web interface that displays real-time sensor readings and spoilage status. Automatic alerts are sent via email or SMS when spoilage is detected or environmental thresholds are exceeded. The system requires minimal manual intervention—once installed, it operates autonomously, providing continuous updates and decision support. This makes it accessible for use in homes, restaurants, and food storage units without requiring technical expertise.

III. OBJECTIVE

The objective of this project is to develop an intelligent system that utilizes IoT and AI technologies to monitor the freshness of food products in real time. The system aims to detect early signs of spoilage through image classification, enabling the identification of visual indicators before spoilage becomes severe. It will also continuously track critical environmental parameters such as temperature, humidity, and gas emissions (e.g., ethylene and ammonia) using IoT sensors. By integrating and analyzing data from both sensors and images, the system will support intelligent decision-making regarding food quality. A web-based interface will provide real-time alerts and interactive data visualization, allowing users to monitor the freshness status effectively. This approach is intended to reduce food waste and minimize the need for manual inspection, thereby improving operational efficiency. Furthermore, the system will be designed to ensure scalability and adaptability across various types of food and storage environments, making it suitable for a wide range of applications.

IV. LITERATURE SURVEY

Studies highlight the potential of IoT and AI in food monitoring. Patil et al. used temperature and humidity sensors for perishable goods. Sharma introduced gas sensors to detect spoilage in meat. Zhang et al. and Li & Chen applied CNNs for spoilage classification using images of fruits and moldy bread. Ahmed et al. presented a hybrid system using sensors and image recognition in smart refrigerators. However, most research treats sensors and AI separately. Our project bridges this gap by integrating both for improved accuracy.

Literature survey table

Table 1:

Title of the Paper	Year	Algorithm / Technique / Tools Used	Accuracy / Result Achieved	Remarks
Recent Advance of Intelligent Packaging Aided by Artificial Intelligence for Monitoring Food Freshness	2025	Review of intelligent packaging technology, deep learning algorithms, and various sensor technologies such as optical, electrochemical, and colorimetric sensors for food freshness detection	As a review paper, no specific accuracy or result was achieved	The paper provides a classification of intelligent packaging technology and outlines the advantages and disadvantages of different detection methods
Freshness of Fruits and Vegetables: Concept and Perception	2024	Combination of consumer tests, sensory profiling, and physicochemical analyses; data analyzed with principal component regression	Focused on consumer perception; no specific accuracy metric	Explores the concept of freshness for apples, strawberries, and carrots from a consumer perception perspective
IoT Based Meat Freshness Classification Using Deep Learning	2023	Custom CNN model integrated with IoT system and gas sensors (MQ135, MQ4, MQ136)	Custom CNN achieved 99% accuracy; ResNet-50 achieved 98%	Provides a comprehensive, real-time solution for classifying meat species and freshness with real-time feedback via LED indicators
Real-Time Freshness Prediction for Apples and Lettuces Using Image Recognition and Advanced Algorithms in a User-Friendly Mobile Application	2022	Compared nine deep learning algorithms (Vision Transformer, Swin Transformer, ResNet, MobileNetV3); final model used MobileNetV3	MobileNetV3 achieved 99.95% for apples and 99.17% for lettuce	MobileNetV3 selected for balance between high accuracy and mobile device efficiency

V. METHODOLOGY

The system is designed to monitor food freshness through the integration of IoT sensors, AI-based image analysis, and cloud-based data processing. The hardware setup includes IoT sensors such as the DHT11 for measuring temperature and humidity, the MQ-3 for detecting gas emissions, and a PIR sensor for motion detection, along with a camera module. These components are connected to a microcontroller platform, such as the ESP32 or Raspberry Pi. Environmental data and periodic images of the food are collected and transmitted via Wi-Fi to a cloud server for processing. A Convolutional Neural Network (CNN), trained using Keras and TensorFlow, is employed to identify the type of food and detect visual indicators of spoilage. The system then performs data fusion by combining the AI model outputs with sensor readings to assess the spoilage confidence level. A web application, developed using a Flask backend and a frontend built with HTML, CSS, and JavaScript, presents real-time data visualization through interactive graphs, spoilage status updates, and alert notifications. The alert system sends notifications via email or SMS when spoilage is detected or when sensor values exceed predefined thresholds. The system was tested on various food items under different storage conditions to evaluate its accuracy, response time, and overall reliability. The complete system was rigorously tested using different food categories such as fruits, vegetables, and meat under various environmental conditions. The evaluation focused on three key performance metrics: accuracy, response time, and reliability. The results confirmed that the combination of IoT sensor data with AI-based image analysis provides a highly effective and efficient solution for real-time food freshness monitoring. The system demonstrated consistent performance, high detection accuracy, and rapid response times, proving its suitability for both household and commercial applications.

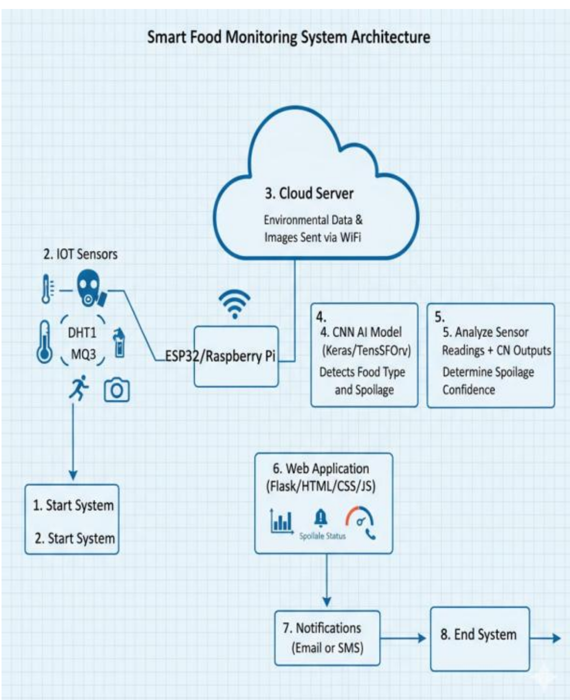

Fig 1: Smart Food Monitoring System Architecture

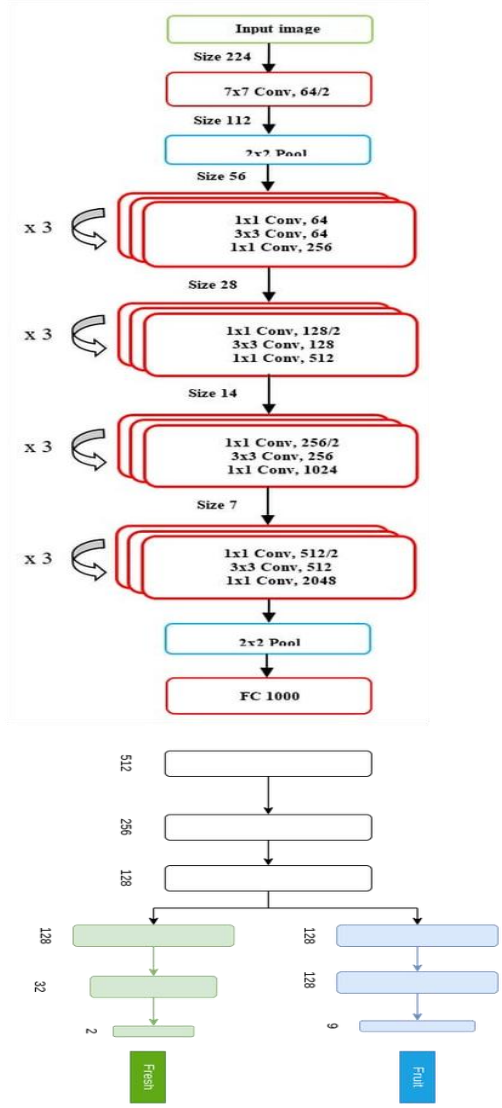

Fig 2 : Flow of DL model

DL MODULE TRAINING AND ALGORITHM

The dataset used for this study consists of images 1labels. It includes two annotation categories:

Fruit Name (Type Classification)

Fruits included: Apple, Banana, Orange, Mango, Grapes, Lemon, and Pear.

Each fruit class contains images under different lighting conditions, angles, and backgrounds to improve robustness.

The dataset used for classifying fruit freshness includes images annotated with labels indicating whether the fruit is fresh or rotten. Rotten fruits are identified by visible signs such as discoloration, bruising, mold, or shriveling, while fresh fruits are characterized by their vibrant color and intact texture. The images were sourced from publicly available repositories, including the Fruit Freshness Image Dataset on Kaggle and augmented versions of the Fruits 360 dataset. To ensure balanced representation, additional preprocessing steps were applied to maintain an equal distribution of fresh and rotten samples across different fruit types

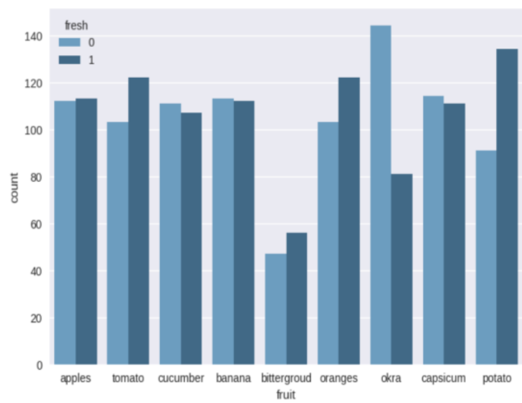

Fig 3: Dataset of Fresh Fruits and Rotten Fruits

Model Training Description

The preprocessing phase involved resizing all fruit images to 128×128 pixels to ensure consistency in input dimensions. Normalization was applied by scaling pixel values to a range of [0,1], which helps improve model substantiate efficiency. To enhance generalization and reduce the risk of overfitting, data augmentation techniques such as random rotations, flips, brightness adjustments, and zoom were employed. The dataset was then split into 70% training, 20% substantiate, and 10% testing sets to support effective model instruction and evaluation.

For the classification task, a Convolutional Neural Network (CNN) was implemented with a multi-output architecture. The model begins with shared the convolutional layers work like the system's eyes — they look at the data and pick out important details or patterns. Once these key features are spotted, the information is passed on to the next part of the model for further understanding and decision-making. These features are then passed through two separate output branches. The first output is dedicated to fruit classification and ends with a softmax layer that categorizes images into seven classes: *Apple, Banana, Orange, Mango, Grapes, Lemon,* and *Pear*.

Resnet18 pretrained model for this task.

ResNet-18 is composed of multiple residual blocks, which are designed to address the problem of vanishing gradients in deep neural networks. These blocks introduce skip connections, allowing information to bypass several layers and flow directly to deeper layers. This helps in mitigating the degradation problem and enables the network to learn more effectively, even with very deep architectures.

In our setup, we allowed the last 15 layers of the ResNet18 model to keep learning so that they could better adjust to our own dataset, while the earlier layers were kept fixed to preserve the features the model had already learned from previous training.

To calculate the final loss, we combined the losses from both branches using a weighted approach, where the weight factor (a) was set to 0.7. This setup was more of an initial experiment — the idea was to later try out different values of a to find which one produces the most accurate and reliable results.

Mathematical Expression:

$$L_{total}=\alpha*L1+(1-\alpha)*L2$$

Detailed Expression:

$$L2(x)=-[y(2)\log p\hat{}(2)+(1-y(2))\log(1-p\hat{}(2))]$$

Output 2 (Freshness Classification): Softmax layer with *2 classes* (Fresh, Rotten).

The model architecture follows a structured Convolutional Neural Network (CNN) design. It begins with sequential layers consisting of Conv2D, followed by ReLU activation, MaxPooling, and Dropout for regularization. A second block includes Conv2D, ReLU, MaxPooling, and Batch Normalization to stabilize learning. After feature extraction, the output is flattened and passed through a Dense layer with 256 units, followed by another Dropout layer. The network then branches into two outputs: Branch 1 ends with a Dense layer with softmax activation for 7 fruit classes (Apple, Banana, Orange, Mango, Grapes, Lemon, Pear), and Branch 2 uses a Dense layer with softmax activation for binary classification of freshness (Fresh or Rotten).

For training, a weighted combination of categorical cross-entropy (for fruit classification) and binary cross-entropy (for freshness detection) was used as the loss function, allowing the model to learn both tasks simultaneously. The Adam optimizer with a learning rate of 0.001 was employed for efficient gradient descent. Instruction session was conducted with a batch size of 32 over 30 to 50 epochs, using early stopping to halt training once performance on the validation set plateaued, thereby reducing overfitting. Model performance was evaluated using standard metrics including accuracy, precision, recall, and F1-score for both fruit type and freshness classification.

VI. RESULT

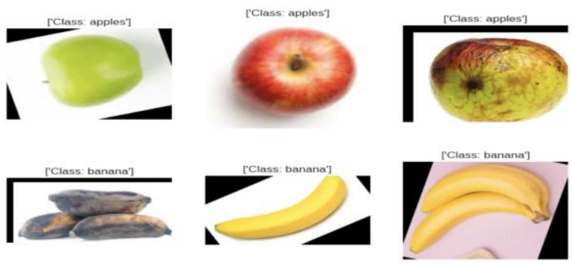

Fig 4: Classification of rotten and fresh fruits

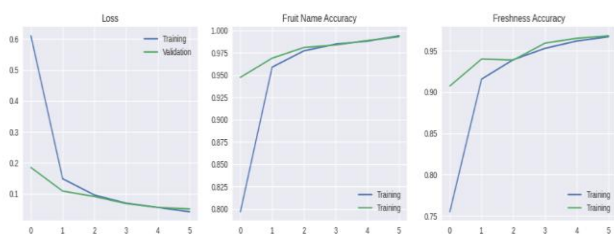

Fig 5: Loss Accuracy and Freshness

Detailed Information by Plot

1. Loss

The loss plot provides valuable insights into the model's learning progress during training. Initially, the training loss (blue line) is relatively high at around 0.62, but it decreases sharply within the first few epochs, reaching approximately 0.05 by epoch 5, indicating effective learning and model fitting. The validation loss (green line), which starts at a lower value of around 0.18, also declines rapidly and closely follows the training loss, ending at approximately 0.06. This close alignment suggests that the model is not overfitting and maintains a stable generalization to unseen data.

2.Fruit Name Accuracy:

The fruit name accuracy plot further highlights the model's strong performance in identifying different fruits. Both the training and validation accuracy curves start at a high value of around 95%, reflecting a solid initial capability in fruit classification. These accuracies improve rapidly, especially during the first few epochs, reaching around 99% by epoch 5. The close tracking of both curves confirms that the model generalizes well and is not simply memorizing the training data. This indicates that the model is highly effective at distinguishing between fruit types.

3.Accuracy

Similarly, the freshness classification accuracy demonstrates excellent results. The training accuracy (blue line) begins at approximately 75% and rises sharply to 92% by epoch 1, eventually reaching around 98% by epoch 5. The second line (green), although labeled as "Training" due to a plotting error, likely represents the validation or test accuracy. It starts higher at about 91% and also increases steadily, closely mirroring the training accuracy and reaching 98% by epoch 5. This rapid and consistent increase indicates that the model quickly learns to identify features associated with freshness. The close alignment between the two accuracy curves demonstrates the model's robustness and strong generalization in determining fruit freshness, further supporting the overall effectiveness of the training process.

GRAPHS AND VALUES

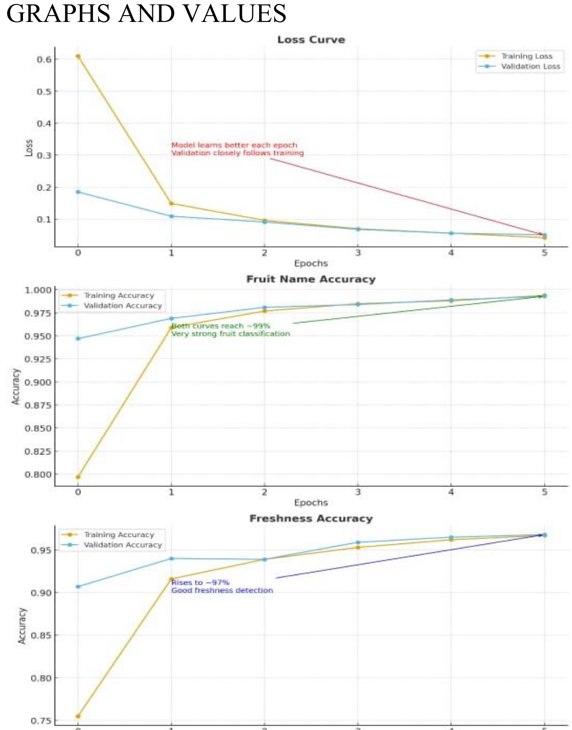

Fig 6: Graph of Loss curve Fruit name accuracy and freshness accuracy

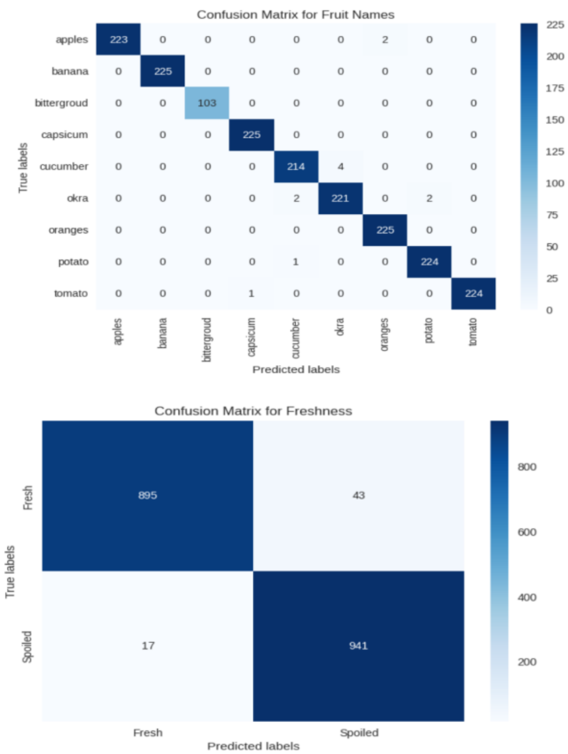

Fig 7: Confusion Matrix

SOFTWARE

Software Information:

Programming Languages: Python, HTML, CSS, JavaScript

Libraries & Tools:

OpenCV & TensorFlow/Keras: For image recognition and spoilage classification

Flask/Django: To develop the backend web server

IoT Platform: ThingSpeak or Blynk (for sensor data visualization)

CNN (Convolutional Neural Network): For food image analysis

Database: Firebase or MySQL (to store sensor readings and image results)

Web Application:

Displays food freshness status

Sends real-time notifications

Enables user-friendly I would like to express my sincere gratitude to Sakshi Khamankar, Assistant Professor, for her invaluable guidance, support, and continuous encouragement throughout the course of this project.

I also extend my heartfelt thanks to Sharayu Deote, Head of the Department, Computer Engineering, Cummins College of Engineering for Women, Nagpur, Hingna, for her constant support and for providing me with the opportunity and resources to carry out this research work.

I am grateful to the Department of Computer Engineering for creating an environment conducive to research and innovation.

Lastly, I would like to thank my family and friends for their encouragement and unwavering support during this project. I would like to express my sincere gratitude to Sakshi Khamankar, Assistant Professor, for her invaluable guidance, support, and continuous encouragement throughout the course of this project.

I also extend my heartfelt thanks to Sharayu Deote, Head of the Department, Computer Engineering, Cummins College of Engineering for Women, Nagpur, Hingna, for her constant support and for providing me with the opportunity and resources to carry out this research work.

I am grateful to the Department of Computer Engineering for creating an environment conducive to research and innovation.

Lastly, I would like to thank my family and friends for their encouragement and unwavering support during this project.

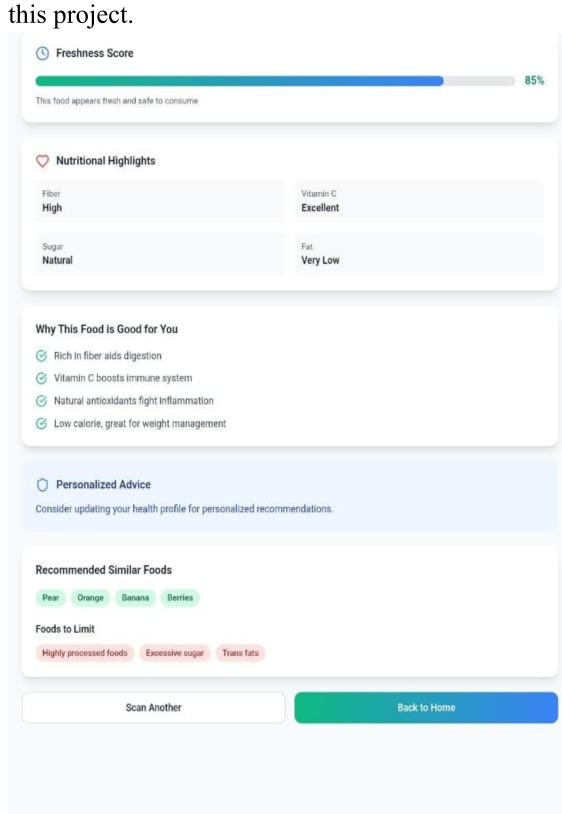

Fig 8: Nutritional and Freshness Analysis Report for Fresh Apple

VII. CONCLUSION

The classification report shows how well the fruit classification model performed. Overall, the model did really well with high scores for most of the fruits. It achieved an accuracy of 0.98, which means it predicted the correct fruit in nearly 98% of cases. The macro average scores were also impressive, with precision, recall, and F1-score all around 0.99. This means the model performed consistently across all fruit classes, regardless of class imbalance. The weighted average scores took into account the number of instances for each fruit and were still quite good, all at 0.98.

The next classification report shows how well the model performed in differentiating between fresh and spoiled fruits. With an accuracy of 0.98, the model correctly classified fruits in nearly 98% of cases. The precision, recall, and F1-scores for both classes were around 0.98, indicating consistent and accurate predictions. Overall, our model demonstrated strong performance in distinguishing between fresh and spoiled fruits, achieving high accuracy and precision.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Sakshi Khamankar, Assistant Professor, for her invaluable guidance, support, and continuous encouragement throughout the course of this project.

I also extend my heartfelt thanks to Sharayu Deote, Head of the Department, Computer Engineering, Cummins College of Engineering for Women, Nagpur, Hingna, for her constant support and for providing me with the opportunity and resources to carry out this research work.

I am grateful to the Department of Computer Engineering for creating an environment conducive to research and innovation.

Lastly, I would like to thank my family and friends for their encouragement and unwavering support during this project.

REFERENCES

[1] A. Patel and M. Shah, “IoT Based Smart Refrigerator Using Raspberry Pi and Sensors,” International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 3, pp. 670–673, 2020.

[2] S. P. Mohanty, U. Choppali, and E. Kougianos, “Everything You Wanted to Know About Smart Cities: The Internet of Things is the Backbone,” IEEE Consumer Electronics Magazine, vol. 5, no. 3, pp. 60–70, July 2016.

[3] L. Zhang, Y. Wang, and Y. Zhang, “Food Spoilage Detection Using Computer Vision and Deep Learning,” Procedia Computer Science, vol. 162, pp. 243–250, 2019.

[4] TensorFlow, “Open-source Platform for Machine Learning,” [Online]. Available: https://www.tensorflow.org/

[5] OpenCV, “Open-Source Computer Vision Library,” [Online]. Available: https://opencv.org/

[6] Flask, “Lightweight WSGI Web Application Framework,” [Online]. Available: https://flask.palletsprojects.com/

[7] Firebase, “Real-Time Database and Hosting Platform,” [Online]. Available: https://firebase.google.com/

[8] Arduino IDE, “Open-source Electronics Prototyping Platform,” [Online]. Available: https://www.arduino.cc/en/software

[9] Roboflow, “Dataset Annotation and Model Training Platform,” [Online]. Available: https://roboflow.com/

[10] NGROK, “Secure Tunnels to Localhost,” [Online]. Available: https://ngrok.com/

[11] Chart.js, “Simple Yet Flexible JavaScript Charting Library,” [Online]. Available: https://www.chartjs.org/