International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 08 | Aug - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Automatic Fruit Freshness Detection Using Deep Learning

Varun Kumar M V[1], Prof.Sandarsh Gowda M M[2]

[1]Student, Department of MCA, Bangalore Institute of Technology, Bangalore, India

[2]Assistant Professor, Department of MCA, Bangalore Institute of Technology, Bangalore, India

---***---

Abstract

Keeping fruits fresh is an important consideration in food safety, nutrition, and consumer satisfaction. Conventional methods of detecting freshness—basically manual inspection—are usually inconsistent, subjective, and not wellsuited for large-scale use. This paper introduces an AI-powered Fruit Freshness Detection System that employs computer vision and machine learning to determine fruit quality in real time. The proposed system takes multi-angle fruit images and classifies the freshness levels while at the same time giving nutritional information. By integrating deep learning (CNN models), image processing (OpenCV), and language-based nutritional analysis (Gemini model), the system provides a strong, scalable, and easy-to-use solution. Rigorous testing confirmed its effectiveness in identifying fruit type, measuring decay, and providing accurate results with fewer error rates. The results underscore the capability of AI-based systems to support supply chain quality checking, reduce food loss, and educate consumers on improved dietary options.

Keywords

Fruit freshness inspection, computer vision, deep learning, convolutional neural networks, artificial intelligence, food quality inspection, nutritional analysis.

1. INTRODUCTION

Fruits are a critical source of necessary vitamins, minerals, and dietary fiber in human diets. Unfortunately, they tend to be very perishable, leading to huge post-harvest losses, which are problems to producers, distributors, and consumers. Fruit freshness maintenance ensures nutritional value, reduces food waste, and economic losses.

Traditional means of determining fruit freshness are generally based on manual inspection wherein retailers and consumers assess quality in terms of visual look, odor, and feel. Simple as these are, they tend to be extremely subjective and vulnerable to human error. Moreover, large-scale agricultural and retailing operations demand automatic, reliable, and scalable quality inspection solutions.

Recent improvements in computer vision and artificial intelligence (AI) have made possible new methods of monitoring the quality of food. By using deep learning models learned on images of fruits, automated systems can identify faint patterns of spoilage that are not perceptible to the human eye. In addition, the coupling of nutritional databases makes such systems value-added, enabling consumers to make healthier food choices.

This study suggests a web-based Fruit Freshness Detection System that integrates image processing, convolutional neural networks (CNNs), and AI-driven analysis to assess fruit quality objectively. In contrast to previous solutions, the system takes multi-angle pictures, transforms them into collages, and delivers real-time freshness ratings along with nutritional information. Its accessibility, scalability, and user-friendliness are prioritized in its design, making it usable throughout the food supply chain—from farm to retail outlets to domestic consumers.

2. LITERATURE SURVEY

Current research identifies several methods of fruit quality measurement, each with advantages and disadvantages: Manual Inspection: The most prevalent method still, but unreliable based on individual expertise and nonscalability. Electronic Sensors: There are systems that measure ethylene gas or other substances, but these involve costly hardware and do not offer thorough freshness analysis.

Basic Computer Vision Models: Early solutions were greatly dependent on the analysis of single-angle images and features based on color, which caused low accuracy and high error rates. Chemical Testing: Chemical analysis through laboratories provides accuracy but is destructive, time-consuming, and unsuitable for use at the retail level.

Modern breakthroughs in deep learning and multi-angle imaging have greatly enhanced detection accuracy. Convolutional neural networks (CNNs) have also been generally applied in agricultural image classification and have performed favorably to identify fruit types and detect decay. In addition, integration with web platforms and cloud storage increases the system's accessibility and scalability.

Although these systems have improved, the majority of them do not have complete solutions that merge freshness detection, nutritional data, user interfaces, and historical trend analysis using data storage. This research bridges these gaps by merging CNN-based image classification with AI-powered nutrition analysis into a web-based system

3. EXISTING SYSTEM

Current Method of Detecting Freshness of Fruits

Presently used fruit freshness detection techniques are based on classical methods with some disadvantages: Visual Inspection – Fruits are evaluated on the basis of appearance, feel, and odor.

Disadvantages: Highly subjective, variable, time-consuming, and not feasible for commercial applications. Electronic Sensors – Gadgets detect gases such as ethylene emitted during the process of ripening.

Disadvantages: Costly, calibration required, and unable to give a comprehensive freshness evaluation.

Basic Computer Vision Approaches – Early systems used simple image analysis based on color or texture. Drawbacks: Limited accuracy, single-angle views, and high error rates.

Chemical Testing – Laboratory methods measure freshnessrelated compounds.

Drawbacks: Destructive, time-consuming, costly, and impractical for consumers or retailers.

Common Issues Across Existing Systems:
Focus only on one parameter (not comprehensive).

Lack of scalability and accessibility.

No integration with AI or nutritional databases.

Poor user expeience and limited real-time usability**.**

4.Proposed System

The suggested Fruit Freshness Detection System utilizes computer vision, web technologies, and artificial intelligence to counter the limitation of current techniques. Five to six images of a fruit from multiple directions are permitted for upload by the users. The uploaded images are then processed through OpenCV for generating a collage so that each face of the fruit is examined. A convolutional neural network (CNN) is used to recognize the fruit type and find patterns of decay. From this analysis, the system computes a freshness score in percentage and classifies the fruit as Fresh, Moderately Fresh, or Spoiled.

Aside from detecting freshness, the system incorporates the Gemini 2.0 model to predict and compute in-depth nutritional data for the recognized fruit. The output is shown to users in real time via an easy-to-use web interface. The frontend is implemented in React.js and Tailwind CSS to provide device responsiveness on desktops and smartphones. Firebase provides secure data storage for users to monitor past analyses and view freshness patterns over a period of time. Chart.js is utilized to display freshness scores and nutritional information through simple visualizations.

The multi-angle method used by the system makes sure that defects hidden on one side of the fruit are not missed. Its ability to process data in real-time gives immediate feedback, which dramatically enhances user experience. The system is architecturally scalable, allowing new fruit types to be easily added in the future. Unlike human inspection, the system does not have any subjectivity and generates consistent and objective results.

The system minimizes food wastage by identifying early signs of spoilage not apparent to the naked eye. It also brings value through the added nutrition information, which makes it possible to make healthier food choices.
Retailers gain from applying the system to quality control of their fruit stock. It is also applicable to consumers, to be used in homes or in markets for assessment of fruit freshness prior to consumption or purchase. Researchers and nutritionists can further use the stored information for research purposes connected to the quality of the food and nutrition. In general, the suggested system offers a complete, dependable, and easyto-use solution for detecting freshness in fruits.

5. IMPLEMENTATION

The deployment of Fruit Freshness Detection System was done in a modular fashion such that every piece of it operates independently but cooperates to provide precise results. The system architecture is client–server-based, with a web-based frontend for user interaction and an AI-powered backend for image processing and nutritional data computation.

1. Frontend Implementation:

The user interface was developed with React.js for its component-centric nature and Tailwind CSS for responsiveness. Users can simply register, login, and upload multiple images of fruits (five to six) via a neat dashboard. Previews of images are shown after upload to ensure correctness of inputs. The frontend speaks to the backend via Axios, performing requests like image upload, analysis initiation, and result fetching. For visualization, Chart.js was implemented to graph freshness trends and nutritional breakdowns, making it possible for users to comprehend results as a graphical presentation.

2. Backend Implementation

The backend was created in Flask (Python), serving as the API layer between machine learning models and the frontend. Flask was selected due to its minimalistic yet robust support for RESTful services. The backend performs various tasks, such as user authentication, image preprocessing, fruit classification, freshness analysis, and generation of nutritional information. The system was made robust with error-handling mechanisms in place to ensure smooth operation under varied situations.

3. Image Processing Module:

Uploaded image of fruits goes through OpenCV preprocessing to maintain uniform quality. The process involves resizing, normalization, and noise elimination. Collage functionality is also supported by OpenCV, where uploaded images are merged into one well-structured layout. This provides the CNN model with a complete view of the fruit, lessening the possibility of overlooking latent defects.

4. AI Model Integration:

A Convolutional Neural Network (CNN) was developed with TensorFlow/Keras and transfer learning from MobileNetV2. The model is trained to recognize fruit types and detect freshness levels based on color changes, surface texture, and patterns of decay. The CNN provides a freshness score (as a percentage) along with fruit classification outcomes. Dropout regularization and fine-tuning methods were used during training to improve accuracy.

Besides the CNN, the system also incorporates Gemini 2.0 Flash, a large-scale language model, to produce comprehensive nutritional data about the identified fruit. The model translates fruit type and freshness information into calorie content, macronutrient composition, and vitamin contents in textual format.

5. Database and Storage:

Firebase Firestore was utilized as the cloud database to store user information, analysis history, and freshness results. Its NoSQL data structure enabled handling of semi-structured data in a flexible manner. Firebase Storage was utilized for storing uploaded fruit images and created collages. Realtime synchronization capabilities help ensure that results are immediately updated for users.

6. Authentication and Security:

Firebase Authentication protects the system by supporting user authentication with email/password or Google login. Role-based access control for administrators was incorporated to allow administrators to control users and system information. Sensitive information is encrypted in transit and at rest to meet security best practices.

7. Deployment and Hosting:

The frontend was hosted on Firebase Hosting, which provides secure global content delivery with fast page loads and SSLenabled secure connections. The backend was on Heroku/PythonAnywhere, providing a reliable Python runtime environment for Flask, TensorFlow, and OpenCV libraries. Environment variables were set securely for API keys and database connections.

8. Testing and Debugging:

During deployment, every module was thoroughly unit tested and integration tested. Jest (for frontend), Pytest (for backend), and Postman (for APIs) were utilized to ensure performance validation. Local testing prior to deployment to prod servers was performed with the Firebase emulator.

9. User Acceptance Testing (UAT):

A similar group of test users used the live system. They tested tasks including registration, image uploads, detection of freshness, and result display. The feedback was gathered to improve the interface and minimize response times. Additions such as mobile optimization and new fruit types were made according to UAT outcomes.

10. Performance Optimization:

To minimize delays, image preprocessing operations were streamlined, and there was also a retry implementation on API calls made to Gemini 2.0 in the event of timeouts. Pagination was included in the results history page to prevent slow loading when large data sets were fetched

6. RESULTS

Fruit Freshness Detection System was thoroughly tested to ensure its functionality, performance, and ease of use. The outcomes prove that the system attained its main goals of automated freshness detection, nutritional analysis, and realtime reporting.

1. Functional Results

The system effectively deployed all basic functions such as user registration, login, image uploads, collage creation, type of fruit classification, freshness level analysis, nutrition facts retrieval, and storing results. Users could do these tasks smoothly on different devices.

2. Fruit Identification Precision and Freshness Determination Accuracy

The Convolutional Neural Network (CNN) was also highly accurate in identifying fruit type and freshness evaluation. Testing involved using familiar fruits like apples, bananas, and oranges, where they were correctly identified with high confidence. The use of multi-angle images eliminated misclassifications by taking into account the concealed defects as well. Visually identical fruits like apples and pears would sometimes trigger slight misclassifications, but were subsequently corrected through retraining the model using other datasets.

3. Nutritional Information Results

The blend of the Gemini 2.0 Flash model included accurate nutritional information for all fruits that were tested. The device offered calories, carbohydrates, fiber, protein, vitamins, and minerals in an orderly manner, with graphical charts included. Customers were pleased with this feature, as it offered more than freshness detection and encouraged responsible dietary choices.

4. Performance Evaluation performance tests indicated that the system was always within response time expectations. Five to six images of fruits (3MB each) were uploaded, processed, and analyzed on average within 12–13 seconds. Collage creation took less than four seconds, whereas freshness classification and nutrition extraction took less than ten seconds. Page loading times were within 2–3 seconds to provide a seamless user experience.

5. Usability and User Experience

User Acceptance Testing (UAT) was performed using 10 test users. Participants all registered successfully, uploaded photos, and viewed results without outside help. Most users found the system very user-friendly, with special praise for the multiangle analysis component and breakdown of nutrients. Some users experienced the occasional sluggish response during high usage, which was overcome through performance enhancements.

6. Error Handling and Stability

The system efficiently managed typical mistakes like incorrect file types, poor-quality images, and incomplete image uploads. Instructive error messages were presented, directing users to make adjustments to their inputs. The backend was stable under average concurrent loads (up to 50 users), although minor delays occurred at peak load, pointing towards future scalability optimization.

7. Quantitative Results

Frontend Unit Tests: 93.3% pass rate

Backend Unit Tests: 92.1% pass rate

Integration Tests: 86.7% pass rate (minor problems with API timeouts)

System Tests: 90% pass rate

Usability Testing: 87.5% success rate

Overall Success Rate: ~91% across all categories

8. Positive Outcomes

Correct detection of freshness levels (Fresh, Moderately Fresh, Spoiled).

Correct classification of fruits for popular varieties. Detailed nutritional information presented along with freshness determination.

Real-time, easy-to-use interface available on web and mobile platforms.

Storage of historical data enabled users to monitor trends in freshness over time.

9. Limitations Observed

Misclassification in visually similar fruit classes.

API timeouts with Gemini integration at high loads. Slower history retrieval when large datasets were stored (fixed with pagination).t

7. CONCLUSION

The Fruit Freshness Detection System effectively proves how computer vision and artificial intelligence can be utilized to address the actual issue of fruit quality evaluation. Through multi-angle imaging, CNN models, and AI-based nutritional analysis, the system makes accurate, objective, and real-time freshness detection possible. The system eliminates the shortcoming of manual examination, minimizes food loss, and provides an added value with nutritional knowledge. With its intuitive web-based interface and scalability, the system is advantageous to both consumers, retailers, and researchers, rendering it an effective and creative solution for contemporary food quality tracking.

Limitations Observed

Misclassification between visually comparable fruit groups.

API timeouts when integrating Gemini under heavy loads.

Slower retrieval of history where large datasets were being stored (fixed with pagination).

8. FUTURE ENHANCEMENT

The Fruit Freshness Detection System can be enhanced further with a number of improvements. Additional fruit and vegetable types may be included to increase its applicability. Predictive models may be added to upgrade the system to estimate the shelf life of fruits that are left. An offline and real-time mobile application may be created for usage. Integration with IoT devices like smart refrigerators can make automated tracking of stored fruits possible. High-level AI models can be included to find contaminants or concealed defects more precisely. Blockchain-based tracking would make the supply chain transparent. Other functionality, such as voice assistant integration, AR-based freshness overlays, and customized nutritional advice, would add user experience value. These enhancements would make the system stronger, scalable, and more effective across consumer, retail, and farming segments.

9. REFERENCES

[1]Ahmad, M., & Singh, S. (2022). Deep learning approaches for fruit quality assessment: A comprehensive review. Journal of Food Engineering, 345, 110–125.

[2]Baker, J. (2023). Computer vision in agricultural applications: Current state and future prospects. Computers and Electronics in Agriculture, 205, 107–119.

[3]Chen, L., & Wong, K. (2022). Multi-angle imaging for enhanced agricultural product quality assessment. Postharvest Biology and Technology, 188, 111–124.

[4]Fernandez, A., & Kumar, P. (2022). Convolutional neural networks for fruit classification and quality assessment. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(8), 1520–1537.

[5]Gupta, V., & Sharma, R. (2023). Computer vision and deep learning for nutrition estimation. Food Research International, 164, 112048.

[6]Johnson, M., & Brown, L. (2022). OpenCV usage in food quality inspection. Computer Vision and Image Understanding, 217, 103390.

[7]Kumar, A., & Desai, V. (2023). Texture analysis for food quality assessment. Computers and Electronics in Agriculture, 201, 106–118.

[8]Lee, J., & Wang, X. (2023). Large language models in nutritional information extraction: Capabilities and limitations.
Journal of Food Composition and Analysis, 118, 104567.

[9]Patel, J., & Rao, T. (2023). Real-time food freshness detection on mobile devices. IEEE Transactions on Mobile Computing, 22(5), 1245–1258.

[10]Sharma, P., & Patel, R. (2022). Transfer learning in fruit quality classification systems. Journal of Food Engineering, 332, 90–105.

[11]Reddy, K., & Mishra, A. (2023). Cloud database integration for food quality assessment applications. International Journal of Cloud Applications and Computing, 13(2), 56–71.

[12]Evans, T., & Zhang, Y. (2023). React.js and Flask: Building responsive full-stack applications. Journal of Software Engineering Practice, 11(2), 45–60.

[13]Das, S., & Patel, R. (2023). Firebase integration in web applications: Best practices and performance considerations.
International Journal of Web Engineering, 15(3), 78–92.

[14]Zhou, Y., & Li, H. (2022). AI-driven food quality monitoring: Challenges and opportunities. Artificial Intelligence in Agriculture, 6, 65–78.

[15]Tanaka, K., & Mori, Y. (2023). Computer vision and deep learning applications in reducing food waste. Sustainable Computing: Informatics and Systems, 38, 100882.