As per UGC guidelines an electronic bar code is provided to secure your paper

International Journal for Modern Trends in Science and Technology
Volume 12, Issue 05, pages 14-19
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue05.html
DOI: https://doi.org/10.5281/zenodo.19613535

Smart System for Identifying and Sorting Rotten Fruits and Vegetables Using Transfer Learning

Ch. Harshitha, D. Abhinava Sai, G. Vrushabha Vahaneswari, G. Uma Lakshmi, I. Vinaya Sri Durga

Department of Computer Science and Engineering (AI & ML), Sir C R Reddy College of Engineering, Eluru, Andhra Pradesh, India

To Cite this Article
Ch. Harshitha, D. Abhinava Sai, G. Vrushabha Vahaneswari, G. Uma Lakshmi & I. Vinaya Sri Durga (2026). Smart System for Identifying and Sorting Rotten Fruits and Vegetables Using Transfer Learning. International Journal for Modern Trends in Science and Technology, 12(05), 14-19. https://doi.org/10.5281/zenodo.19613535

Article Info
Received: 28 March 2026; Revised: 24 April 2026; Accepted: 26 April 2026.

KEYWORDS

Transfer Learning, EfficientNetB0, ResNet50, MobileNetV2, Fruit Classification, Shelf-Life Prediction, Deep Learning, CNN, Grad-CAM, Streamlit.

ABSTRACT

The quality and freshness of fruits and vegetables play a crucial role in food safety, supply chain efficiency, and waste reduction. Traditional methods of freshness assessment rely on manual inspection, which is subjective and often inaccurate. Existing AI-based systems mainly focus on binary classification (fresh or rotten), which is insufficient for real-world decision-making. This project presents a Smart System for Identifying and Sorting Rotten Fruits and Vegetables using Transfer Learning, which extends beyond classification to provide a complete shelf-life intelligence pipeline. The system integrates three deep learning models: a MobileNetV2-based classifier for automatic produce identification, a ResNet50-based regression model for predicting freshness score, and an EfficientNetB0-based regression model for estimating days remaining before spoilage. The system processes an uploaded image and outputs produce type, freshness condition, freshness score, and estimated shelf life along with actionable recommendations. Grad-CAM is used for explainability to visualize model focus regions. The system is deployed using Streamlit, providing a user-friendly interface for real-time predictions.

1. INTRODUCTION

In today's world, food waste is a major global issue, especially in perishable items like fruits and vegetables. A significant portion of this waste occurs due to improper assessment of freshness and shelf life. Traditional methods depend on manual inspection, which is subjective, inconsistent, and inefficient.

With advancements in machine learning and deep learning, automated systems can now analyze images and extract meaningful insights. However, most existing systems only classify produce as fresh or rotten, lacking detailed insights such as how fresh the item is or how long it will last.

This project aims to develop a smart system that:

- Automatically identifies the type of fruit or vegetable
- Predicts its freshness level
- Estimates remaining shelf life
- Provides actionable recommendations

The system is designed for:

- Retail stores
- Supply chain management
- Consumers
- Agricultural applications

2. LITERATURE REVIEW

The application of deep learning to fruit and vegetable quality assessment has evolved from handcrafted feature extraction to end-to-end convolutional learning. Early approaches used color histograms, texture descriptors, and support vector machines for binary freshness classification, achieving 80–90% accuracy on small controlled datasets [1][2].

With the advent of deep CNNs, Fahad et al. [3] applied VGG-16 and YOLO for three-class freshness categorization achieving 82–84% accuracy. Mukhiddinov et al. [4] used improved YOLOv4 for real-time fresh/rotten detection. The most directly relevant work, published at ICIMIA-2025 [5], applied transfer learning with ResNet50, MobileNetV2, and EfficientNetB0 on a 12,000-image dataset achieving up to 99.29% classification accuracy. All these studies output categorical labels rather than quantified shelf-life predictions.

Koyama et al. [6] approached freshness as a regression problem for spinach, predicting sensory scores using color features with SVM — one of the few regression-based studies in this domain. Yuan et al. [7] integrated CNNs with BiLSTM for time-series freshness tracking. However, continuous regression across 12 produce types with simultaneous score and days prediction remains unexplored in published literature.

Dosovitskiy et al. [8] introduced Vision Transformers (ViT), demonstrating competitive performance with CNNs on large-scale image classification. Several studies have compared ViT and CNN performance in medical and satellite imaging domains, consistently finding that ViT requires significantly larger datasets than CNNs to be competitive. To the best of our knowledge, no prior study has compared ViT versus CNN architectures specifically for agricultural shelf-life regression — a comparison this paper provides.

Selvaraju et al. [9] proposed Grad-CAM for visual explanation of CNN decisions, which has been applied in medical imaging to validate model attention on pathological regions. Application of Grad-CAM to produce freshness models to validate biological correctness of attention regions has not been previously reported.

3. METHODOLOGY

The proposed system follows a structured pipeline consisting of multiple stages, including data collection, preprocessing, model training, and prediction.

3.1 Data Collection

The dataset used in this project is collected from multiple publicly available sources, primarily Kaggle datasets(4 datasets are used). It includes images of various fruits and vegetables under different conditions such as fresh, moderately fresh, and rotten. The dataset covers multiple produce types to ensure diversity and robustness.

3.2 Data Preprocessing

Since the images originate from different datasets, preprocessing plays a crucial role in ensuring consistency before feeding data into the models.

Step 1: Image Format Standardisation — All uploaded images, regardless of their original file format (JPG, PNG, WEBP, BMP), are First, all images are converted into a standard RGB format to maintain uniformity across inputs. This step eliminates issues caused by grayscale or multi-channel variations.

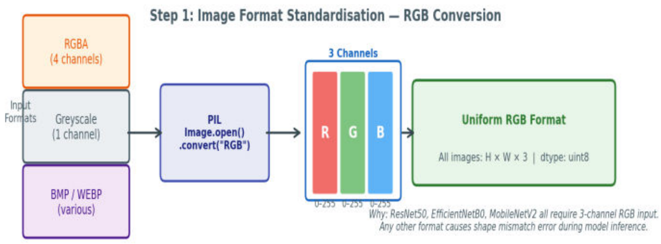


Fig 1: RGB format standardisation — all input formats converted to uniform 3-channel RGB

Step 2: Resizing to 224×224 Pixels — Each image is resized to 224 × 224 pixels, which matches the input requirements of the pretrained deep learning models used in this system. This resizing ensures compatibility while preserving important visual features.

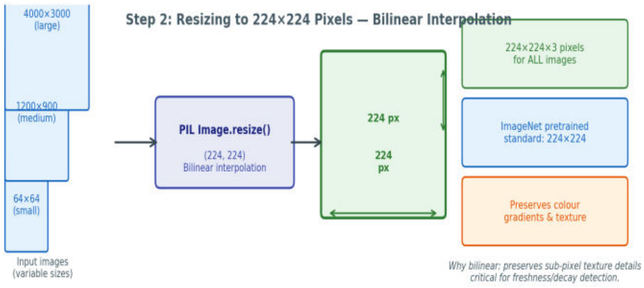


Fig 2: Bilinear resizing to 224×224 px — required by all three pretrained backbones

Step 3: Pixel Value Normalisation — After resizing, all Pixel values are then converted into floating-point format, maintaining consistency with the training configuration. Unlike typical normalization practices, the pixel range is preserved to align with how the models were originally trained.

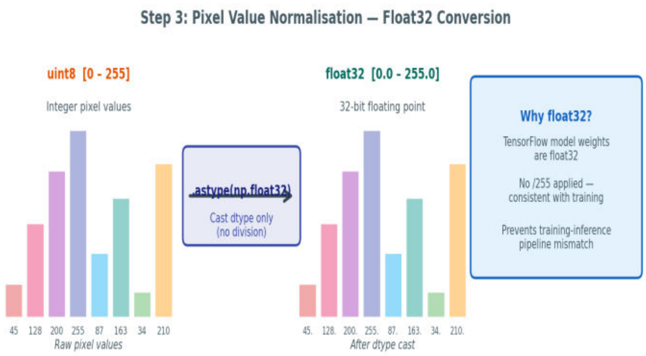


Fig 3: Float32 pixel conversion — matching training and deployment preprocessing exactly

Step 4: Data Augmentation (Training Only) — To improve generalization, data augmentation techniques are applied during training. These include random horizontal and vertical flips, as well as slight brightness adjustments. Such transformations help the model learn robust features and reduce overfitting.

Fig 4: Data augmentation (training only) — flip and brightness jitter

Step 5: Batch Construction and Shuffling — Images are processed in batches using custom data generators, allowing efficient memory usage during training. Additionally, corrupted or missing files are handled gracefully by substituting them with placeholder data, ensuring uninterrupted model execution.

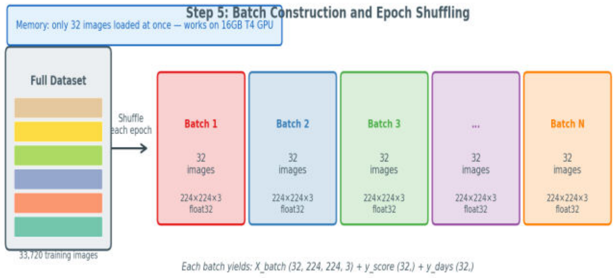


Fig 5: Batch construction with epoch-level shuffling — 32 images/batch

Step 6: Handling Missing or Corrupted Images — Since the data was gathered from multiple Kaggle sources, not every file was perfectly usable. A few images turned out to be missing, damaged, or unreadable after being transferred to the working environment. Instead of letting these issues interrupt the training process, a simple safeguard was added.

Each time an image is loaded, the process is wrapped in a try-except block. If an error occurs (for example, if the file cannot be opened), the system replaces that image with a blank NumPy array of size 224 × 224 × 3. This way, the batch size stays consistent and training can continue without crashing due to a few faulty files.

In addition, all file paths were checked beforehand using pandas to make sure they actually existed. After removing invalid entries, the dataset was reduced from roughly 51,000 images to about 48,000 clean and usable samples for training and evaluation.

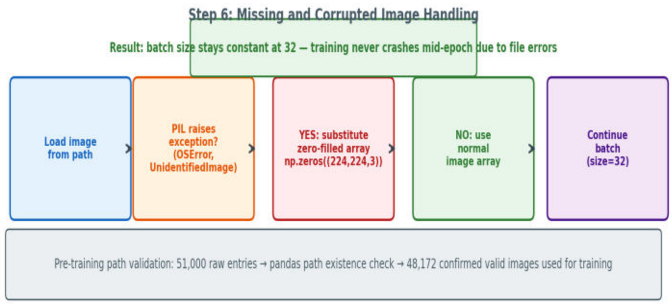


Fig 6: Corrupted image handling — try-except fallback, training never crashes

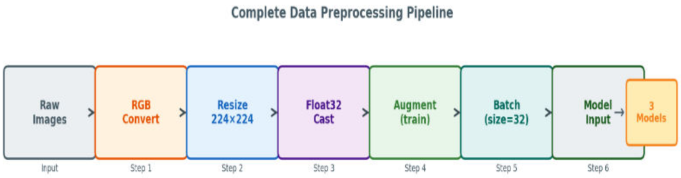


Fig 7: Complete 6-step preprocessing pipeline overview — raw Kaggle images to model-ready tensors

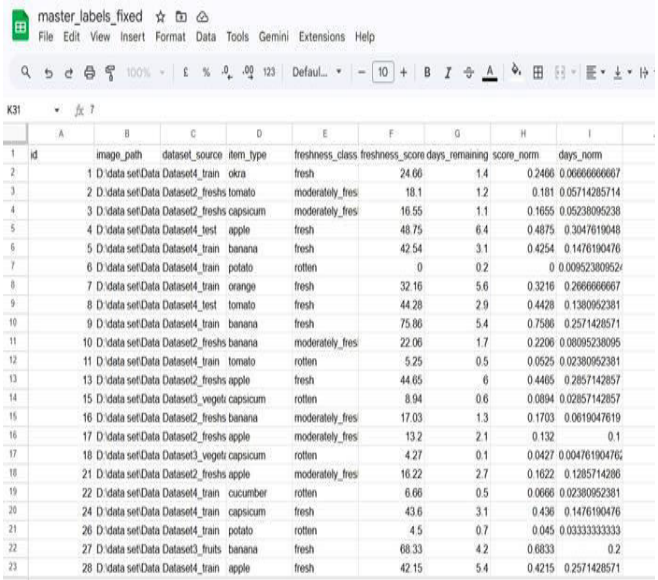
Taken together, these six preprocessing steps create a consistent and reliable pipeline that transforms raw images from different Kaggle datasets into a format suitable for model input. Since the images originally come from varied sources, this step helps bring everything to a common standard before training.

The same preprocessing logic is used both during training (in the notebook) and in the final Streamlit application. Keeping this process consistent is important, because it ensures that the model receives inputs in the same format during real-world use as it did during training. Without this alignment, the model's performance could become unpredictable.

3.3 Label Engineering Methodology

A key contribution of this work is the systematic conversion of categorical freshness labels into continuous regression targets. Initial experiments using percentage-based ranges (moderately_fresh = 25–64% of total shelf life) produced systematic errors for long-shelf-life items: orange moderately_fresh was assigned up to 8.96 days despite being clearly visually deteriorated, causing the model to learn incorrectly calibrated predictions.

This was corrected by establishing USDA-verified absolute day ranges for each produce type and condition based on room-temperature storage guidelines. For example, orange: fresh=5–7 days, moderately_fresh=2–4 days, rotten=0–1 days. Freshness score is computed as (days/total_shelf_life) × 100 with Gaussian noise (σ=3) to prevent discrete label boundaries. This methodology improved model performance by 29–87% across all architectures.


master_labels_fixed

File Edit View Insert Format Data Tools Gemini Extensions Help

K31 fx 7

	id	image_path	dataset_source	item_type	freshness_class	freshness_score	days_remaining	score_norm	days_norm
2	1	D:\data set\Data	Dataset4_train	okra	fresh	24.66	1.4	0.2466	0.06666666667
3	2	D:\data set\Data	Dataset2_freshs	tomato	moderately_fres	18.1	1.2	0.181	0.05714285714
4	3	D:\data set\Data	Dataset2_freshs	capsicum	moderately_fres	16.55	1.1	0.1655	0.05238095238
5	4	D:\data set\Data	Dataset4_test	apple	fresh	48.75	6.4	0.4875	0.3047619048
6	5	D:\data set\Data	Dataset4_train	banana	fresh	42.54	3.1	0.4254	0.1476190476
7	6	D:\data set\Data	Dataset4_train	potato	rotten	0	0.2	0	0.00952380952
8	7	D:\data set\Data	Dataset4_train	orange	fresh	32.16	5.6	0.3216	0.2666666667
9	8	D:\data set\Data	Dataset4_test	tomato	fresh	44.28	2.9	0.4428	0.1380952381
10	9	D:\data set\Data	Dataset4_train	banana	fresh	75.86	5.4	0.7586	0.2571428571
11	10	D:\data set\Data	Dataset2_freshs	banana	moderately_fres	22.06	1.7	0.2206	0.08095238095
12	11	D:\data set\Data	Dataset4_train	tomato	rotten	5.25	0.5	0.0525	0.02380952381
13	13	D:\data set\Data	Dataset2_freshs	apple	fresh	44.65	6	0.4465	0.2857142857
14	15	D:\data set\Data	Dataset3_veget	capsicum	rotten	8.94	0.6	0.0894	0.02857142857
15	16	D:\data set\Data	Dataset2_freshs	banana	moderately_fres	17.03	1.3	0.1703	0.0619047619
16	17	D:\data set\Data	Dataset2_freshs	apple	moderately_fres	13.2	2.1	0.132	0.1
17	18	D:\data set\Data	Dataset3_veget	capsicum	rotten	4.27	0.1	0.0427	0.00476190476
18	21	D:\data set\Data	Dataset2_freshs	apple	moderately_fres	16.22	2.7	0.1622	0.1285714286
19	22	D:\data set\Data	Dataset4_train	cucumber	rotten	6.66	0.5	0.0666	0.02380952381
20	24	D:\data set\Data	Dataset4_train	capsicum	fresh	43.6	3.1	0.436	0.1476190476
21	26	D:\data set\Data	Dataset4_train	potato	rotten	4.5	0.7	0.045	0.03333333333
22	27	D:\data set\Data	Dataset3_fruits	banana	fresh	68.33	4.2	0.6833	0.2
23	28	D:\data set\Data	Dataset4_train	apple	fresh	42.15	5.4	0.4215	0.2571428571

Fig 8: master labled data

3.4 Model Architecture

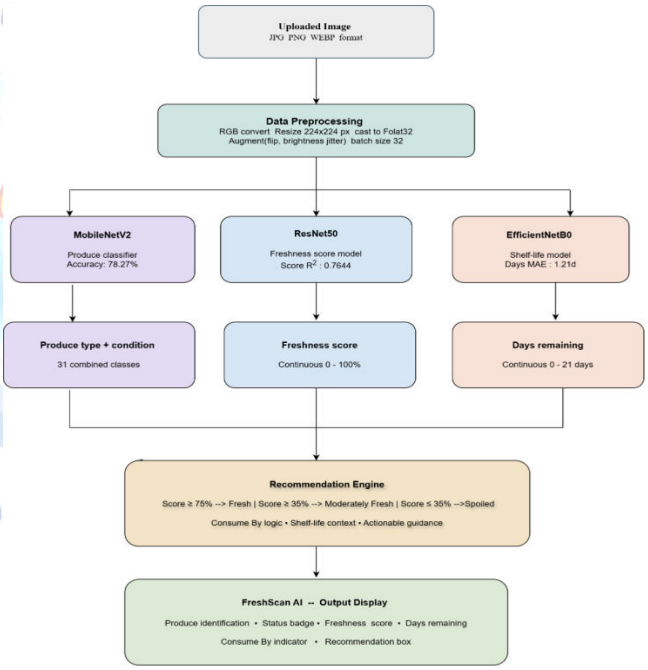


Fig 9: Architecture

The system uses three deep learning models:

1. MobileNetV2 (Classification Model)

- Used for identifying the type of fruit or vegetable
- Lightweight and efficient
- Suitable for real-time applications

2. ResNet50 (Freshness Score Model)

- Used for predicting a continuous freshness score (0–100%)
- Deep architecture helps in capturing complex features

3. EfficientNetB0 (Shelf-Life Prediction Model)

- Used for predicting days remaining before spoilage
- Optimized for performance and efficiency

3.5 Transfer Learning

All models use transfer learning, where pre-trained weights from ImageNet are used. The lower layers are frozen, and the upper layers are fine-tuned using the dataset. This approach reduces training time and improves accuracy.

3.6 Grad-CAM Analysis

Grad-CAM analysis on EfficientNetB0 confirmed biologically correct spatial attention: the model focused on the tomato surface for fresh produce prediction, on the banana stem region (a key ripeness indicator) for fresh banana assessment, and precisely on dark rot spots for rotten potato classification. These results validate that models are learning relevant visual features rather than memorising background characteristics. ResNet50 showed weaker spatial localization due to greater feature abstraction in the final convolutional block, though earlier-layer Grad-CAM (conv4) improved localization quality.

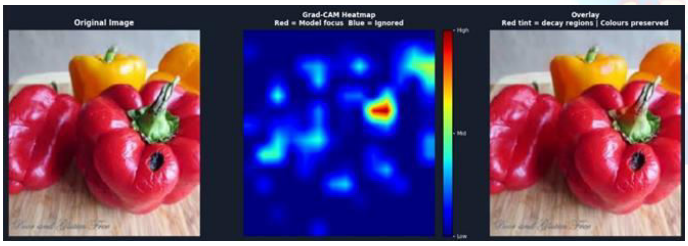


Fig 10: Grad-Cam Anaysis

3.7 System Workflow

- User uploads an image
- Image is preprocessed
- MobileNetV2 identifies produce type
- ResNet50 predicts freshness score
- EfficientNetB0 predicts days remaining
- Results are displayed to the user

4. RESULTS AND OUTPUTS

This section presents the results generated by the Smart System for Identifying and Sorting Rotten Fruits and Vegetables using Transfer Learning.

Fig 11: User Interface

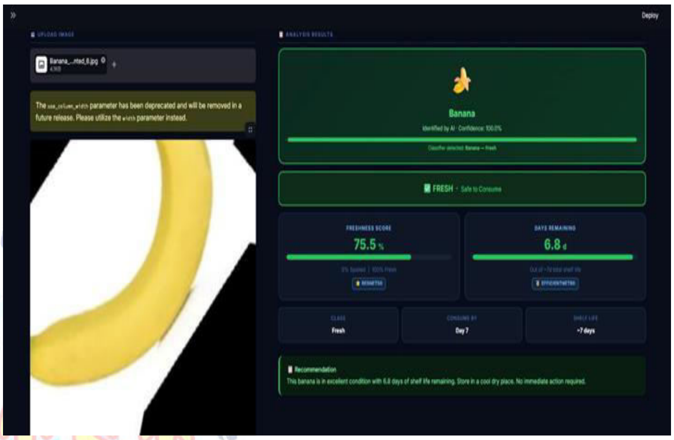


Fig 12: A Fresh Fruit is Predicted along with its Freshness score and Shelf_life

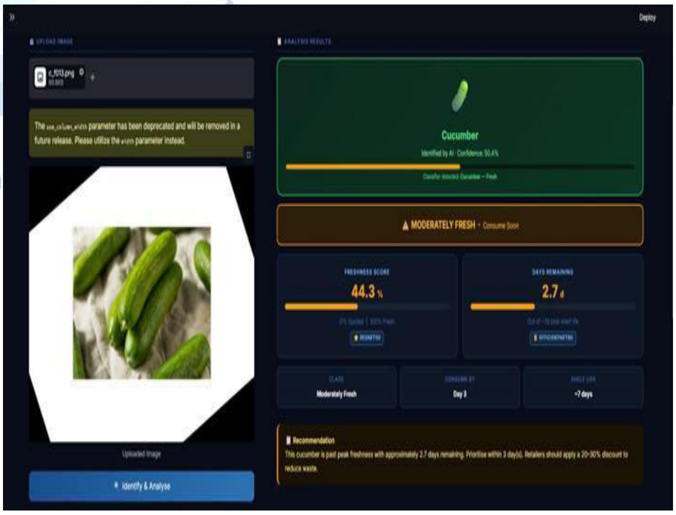


Fig 13: A moderately Fresh Vegetable is Predicted along with its Freshness score and Shelf_life

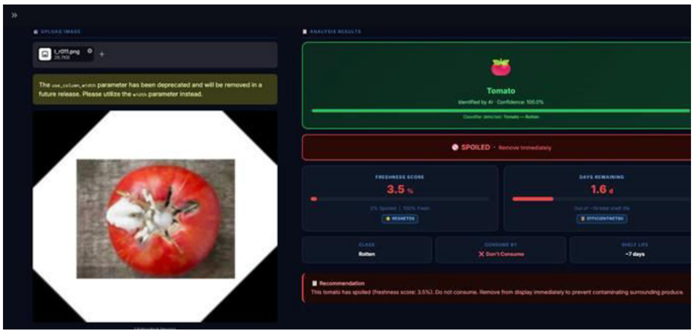


Fig 14: A Rotten Vegetable is Predicted along with its Freshness score and Shelf_life

6. CONCLUSION

This work presents a system that combines three transfer learning models trained on a dataset of over 48,000 images covering 12 different types of fruits and vegetables collected from multiple Kaggle sources. The MobileNetV2 model is used to identify the type of produce and its condition, achieving an overall accuracy of around 78%, with particularly strong performance on clearly fresh and clearly spoiled items. Alongside this, a ResNet50-based model estimates a continuous freshness score, while an EfficientNetB0 model predicts how many days remain before the item is likely to spoil.

An important part of this study was the way the labels were designed. Instead of relying on simple percentage-based assumptions, shelf-life values were assigned using reference guidelines, which were then converted into continuous targets. This change led to noticeable improvements in model performance across all architectures.

The system is implemented as a Streamlit application called FreshScan AI, making it easy to use without any setup. Once an image is uploaded, the models run in sequence and provide a set of outputs, including the identified produce type, its condition, a freshness score, and an estimate of remaining shelf life. The interface also presents this information in a clear visual format, along with simple recommendations. For example, when the predicted freshness drops below a certain threshold, the system explicitly warns the user not to consume the item.

To better understand how the models arrive at their predictions, Grad-CAM visualization was applied. The results showed that the EfficientNetB0 model focuses on relevant areas such as surface texture in tomatoes, the stem region in bananas, and visible decay spots in potatoes. This indicates that the system is learning meaningful visual patterns rather than relying on irrelevant background details.

Overall, the proposed approach moves beyond basic classification and offers a more practical way to assess produce quality, with potential benefits for consumers, retailers, and supply chain management.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Koyama, K., et al. Predicting sensory evaluation of spinach freshness using machine learning model and digital images. PLOS ONE, 16(3), 2021.

[2] Zhang, B., et al. Computer vision for external quality inspection of fruits and vegetables: principles, developments and applications. Computer Science Review, 2014.

[3] Fahad, L. G., et al. Fruits and vegetables freshness categorization using deep learning. Computers, Materials & Continua, 71(3), 5083–5098, 2022.

[4] Mukhiddinov, M., et al. Improved classification approach for fruits and vegetables freshness based on deep learning. Sensors, 22(21), 8192, 2022.

[5] Subrahmanyam, V., et al. An Innovative Approach for Detecting the Freshness of Fruits and Vegetables Using Transfer Learning Techniques. Proceedings of ICIMIA-2025, IEEE, 2025.

[6] Koyama, K., et al. Predicting sensory evaluation of spinach freshness using machine learning. PLOS ONE, 16(3), e0248798, 2021.

[7] Yuan, Y., et al. An innovative approach to detecting freshness through the integration of CNNs and BiLSTM. Current Research in Food Science, 8, 2024.

[8] Dosovitskiy, A., et al. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. ICLR, 2021.

[9] Selvaraju, R. R., et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization. ICCV, 2017.

[10] Tan, M., & Le, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. ICML, 2019.

[11] He, K., et al. Deep residual learning for image recognition. IEEE CVPR, 2016.

[12] Howard, A., et al. MobileNetV2: Inverted residuals and linear bottlenecks. IEEE CVPR, 2018.

[13] Food and Agriculture Organization (FAO). The State of Food and Agriculture: Moving Forward on Food Loss and Waste Reduction. FAO, Rome, 2019.

[14] USDA and land-grant extension publications. Room-temperature shelf-life data for fresh produce. StopWaste, StillTasty, MSU Extension, UC Davis, UF/IFAS. Accessed 2025–2026.

[15] Oliullah, K., et al. FruVeg_MultiNet: A hybrid deep learning-enabled IoT system for fresh fruit and vegetable identification. Journal of Agriculture and Food Research, 2025.