

A Lightweight Browser-Based CNN for Fresh Produce Freshness Classification Using MobileNetV2

Rovhona Mudau

Department of Computer Science, University of the Western Cape

Jean Frederic Isingizwe Nturambirwe

fisingizwe@uwc.ac.za

eResearch Office, University of the Western Cape <https://orcid.org/0000-0002-1794-7343>

Ikechukwu Kingsley Opara

Department of Computer Science, University of the Western Cape <https://orcid.org/0000-0003-0747-5319>

Clement Nyirenda

Department of Computer Science and eResearch Office, University of the Western Cape
<https://orcid.org/0000-0002-4181-0478>

Research Article

Keywords: Fresh produce, Postharvest quality, Visual Freshness Index , Convolutional neural networks, MobileNetV2

Posted Date: October 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11434553/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

A Lightweight Browser-Based CNN for Fresh Produce Freshness Classification Using MobileNetV2

Rovhona Mudau¹, Jean Frederic Isingizwe Nturambirwe²[0000-0002-1794-7343]★,
Ikechukwu Kingsley Opara¹[0000-0003-0747-5319], and Clement
Nyirenda^{1,2}[0000-0002-4181-0478]

¹ Department of Computer Science, University of the Western Cape, Robert Sobukwe Rd,
Bellville, Cape Town, 7535, South Africa
4225313@myuwc.ac.za, 6509709@myuwc.ac.za

² eResearch Office, University of the Western Cape, Robert Sobukwe Rd, Bellville, Cape Town,
7535, South Africa
fisingizwe@uwc.ac.za, cnyirenda@uwc.ac.za

Abstract. Postharvest losses of fresh produce threaten food security, particularly in resource-constrained supply chains where quality is still assessed by subjective manual inspection. This study presents a two-component system: a MobileNetV2 convolutional neural network that classifies images of produce as fresh or rotten, and a browser-based web application that runs the classifier entirely client-side using TensorFlow.js, without a server or specialised hardware. Using a public dataset of roughly 30,000 labelled images spanning nine produce types, the model achieved 98.46% accuracy on a held-out test set, with a macro-averaged F1-score of 0.985 and an ROC-AUC of 0.9989. The confidence output, presented to users as a Visual Freshness Index (VFI), was well calibrated against the binary label (Expected Calibration Error: 0.0051), making it a dependable indicator of classification confidence. Trained with native preprocessing, the heavier baselines ResNet50, EfficientNetB0, and VGG16 achieved comparable accuracy, so MobileNetV2 was retained for its small footprint and suitability for deployment. The shelf-life and storage suggestions shown by the application are indicative heuristics, not validated model outputs. Its offline, browser-based design makes the system potentially suited to resource-constrained and low-connectivity agricultural settings.

Keywords: Fresh produce · Postharvest quality · Visual Freshness Index · Convolutional neural networks · MobileNetV2

1 Introduction

Global demand for fresh produce has risen with population growth, increasing health awareness, and the drive towards more sustainable food systems [9]. Yet a substantial fraction of produce is lost to inadequate storage and handling, generating both economic losses and environmental harm [8,20]. Roughly 30% of food produced worldwide is wasted, with fruits and vegetables accounting for a disproportionate share owing to

★ Corresponding author.

their perishability [8]. This is paradoxical given that fruit and vegetable intake supplies micronutrients, dietary fibre, and phytochemicals that lower the risk of obesity and chronic disease [6,29], and that the World Health Organisation (WHO) recommends a minimum of 400 g per day to improve population health outcomes [33]. Quality deterioration along the supply chain has compounding costs. Inappropriate temperature and humidity accelerate microbial spoilage [20]. Microbial contamination of fresh produce poses food-safety and economic risks, while bruising and other mechanical damage can reduce market quality, cause downgrading or rejection, and contribute to postharvest losses [1,22]. Against this backdrop, machine learning has delivered measurable gains in several aspects of agriculture, such as crop yield forecasting, automated pest and disease detection, precision irrigation, and soil-nutrient management, with reported efficiency improvements and resource-waste reductions of 15–30% [3,5,11,14,21]. What remains comparatively underserved is an accessible tool that helps frontline stakeholders make better handling and storage decisions without specialised infrastructure.

This study focuses on the limited accessibility of automated visual produce-quality assessment in resource-constrained settings. Although machine-learning approaches can support produce evaluation, their practical adoption may be constrained when assessment depends on specialised sensing equipment, laboratory measurements, or additional sensing and communication infrastructure [19,25,31]. At the same time, existing computer-vision studies commonly emphasise classification performance, while confidence, calibration, and practical deployment receive comparatively less attention [7,23]. Such performance estimates may also be misleading when train and test partitions contain duplicated or closely related observations, motivating the leakage-robustness analysis undertaken in this study. There is therefore a need for a lightweight visual classification approach that can operate on commonly available browser-capable devices while characterising predictive performance beyond headline accuracy. In this study, the problem is deliberately bounded to visual fresh-versus-rotten classification; physical freshness, remaining shelf life, and physicochemical quality are not inferred from the image classifier.

This paper addresses that gap with a two-component system: a MobileNetV2-based Convolutional Neural Network (CNN) binary classifier and a browser-based web application that wraps the classifier for practical, offline-capable deployment. The classifier discriminates between visually fresh and rotten produce from colour, surface texture, and visible blemishes. The application surfaces the classification, a confidence-derived VFI, indicative storage guidance, and an exportable PDF report. In scope, the storage suggestions are heuristic guidance rather than validated forecasts. The VFI is not a new quantity but a presentation device: it rescales the model's fresh-class probability to an intuitive index so that non-specialist users receive a graded indication of confidence rather than a bare binary verdict, which supports triage decisions near the decision boundary. Because the dataset provides only binary fresh/rotten labels, with no paired physicochemical measurements, the index can be calibrated against that label alone; establishing it as a measure of physical freshness would require destructive laboratory assays such as firmness, pH, soluble-solids content, or microbial load, recorded image by image, which lies beyond the scope of this study.

Three gaps motivate this work. First, practical deployment of automated produce-quality assessment can be constrained by dependence on specialised sensing equipment, laboratory measurements, or additional monitoring infrastructure, which increases cost and operational complexity in resource-constrained agricultural contexts [19,25,31]. Image-based assessment offers a lower-infrastructure alternative because it can operate using cameras available on commodity devices, and previous work has demonstrated the feasibility of MobileNetV2-based fruit-freshness classification on mobile platforms [16]. Second, existing computer-vision studies on fruit-quality assessment commonly emphasise classification performance [7,23], while classification accuracy alone does not establish whether predicted probabilities provide reliable confidence estimates [10]. Performance estimates may also be misleading when training and test partitions contain duplicate or closely related observations, because such overlap can inflate apparent generalisation [4]. Third, deployment architecture remains an important practical consideration. Lightweight client-side inference allows computation to take place directly on the user’s device rather than requiring server-side inference for each prediction. TensorFlow.js enables machine-learning models to execute directly within web browsers and supports on-device computation [30]. This browser-based approach therefore reduces dependence on continuous network connectivity during inference and avoids the need for platform-specific application installation. Together, these gaps motivate a lightweight visual classification system that combines reliability-oriented evaluation with browser-based client-side deployment.

This work makes four principal contributions. First, we develop a lightweight MobileNetV2-based classifier for automated fresh-versus-rotten produce classification and evaluate it beyond headline accuracy using class-level performance metrics, discrimination measures, calibration analysis, bootstrap confidence intervals, and a dataset-leakage robustness audit. Second, we derive the VFI from the model’s fresh-class probability and present it explicitly as a calibrated classification-confidence indicator rather than a direct physicochemical measure of freshness. Third, we deploy the trained model as a browser-based TensorFlow.js application that performs inference entirely client-side, without requiring a server, continuous cloud connectivity, or specialised sensing hardware. Fourth, we compare MobileNetV2 with ResNet50, EfficientNetB0, and VGG16 under architecture-specific preprocessing, examining the trade-off between predictive performance, parameter efficiency, and suitability for client-side deployment. Together, these contributions combine reliability-characterised visual classification with an accessible deployment framework for postharvest produce assessment.

2 Machine Learning Approaches in Agriculture

2.1 Statistical Models

Early approaches to produce quality prediction used statistical models that correlated storage conditions with quality outcomes from historical data [14]. Partial Least Squares Regression and Multiple Linear Regression have been used to forecast shelf-life and the evolution of physicochemical attributes such as firmness, pH, and soluble solids content in commodities like apples and tomatoes [2,24]. Multivariate techniques such as Principal Component Analysis combined with Analysis of Variance were used to

track spoilage kinetics under varying temperature and humidity. These methods offered interpretability but relied on manually chosen variables and largely linear assumptions, limiting their ability to capture the highly non-linear dynamics of spoilage, and typically required controlled laboratory measurement, constraining real-world deployment.

2.2 Traditional Machine Learning

Classical machine learning introduced models capable of capturing non-linear structure. Support Vector Machines (SVMs), Decision Trees, Random Forests, and Artificial Neural Networks (ANNs) have been applied to agricultural monitoring for defect detection and produce evaluation [14]. SVMs have been used effectively for non-destructive detection of surface defects and bruising from image and spectral features [19], and ANNs have modelled postharvest changes under varying storage and packaging conditions [26]. These methods handled larger datasets and outperformed statistical baselines, but remained dependent on subjective, time-consuming manual feature engineering.

2.3 Deep Learning for Computer Vision in Agriculture

Deep learning has reshaped agricultural image analysis by learning hierarchical representations directly from raw images, reducing reliance on manually engineered visual features [11,15]. CNN-based methods have consequently become prominent in agricultural computer-vision tasks because they can learn discriminative patterns associated with colour, texture, shape, and visible surface defects [7,11]. MobileNetV2 and NAS-NetMobile have been used to classify banana ripeness across maturity stages [27]. CNNs have achieved strong performance in plant disease diagnosis from leaf and fruit imagery [15], and deep models have distinguished fresh from spoiled produce using visual cues such as colour change, texture degradation, bruising, and mould [34]. Closely related to the present work, Palakodati et al. [23] reported strong fresh/rotten classification accuracy using VGG-based transfer learning, while Singh et al. [28] investigated fresh/rotten fruit classification using deep CNNs and MobileNetV2. The present study builds on this line of work by combining MobileNetV2-based classification with confidence calibration, dataset-leakage analysis, comparison with alternative CNN backbones, and client-side browser deployment. The recent review by Chuquimarca et al. [7] further surveys CNN models for external fruit quality inspection and grading.

2.4 Internet of Things (IoT) Enabled Monitoring

Coupling IoT with machine learning extends monitoring to continuous environmental sensing; temperature, humidity, and gas concentration are analysed to support decisions [32]. Low-cost spectral sensing with regression models has assessed produce condition non-destructively [31], while IoT platforms incorporating temperature, humidity, and gas sensors have been used to monitor fruit ripening and spoilage [25]. Nsunba et al. [18] developed a real-time environmental-monitoring framework for food cold stores in which physical sensors are represented through a Digital Twin architecture, with automated acquisition, decoding, storage, querying, and analysis of temperature, humidity,

and gas measurements. Such systems demonstrate how continuous environmental data can support cold-chain monitoring and provide a foundation for subsequent predictive and machine-learning applications. Supply-chain monitoring systems have similarly been proposed to optimise logistics and reduce food losses and waste [13]. These approaches provide valuable environmental context but require additional hardware, sensor calibration, data-management infrastructure, and communication mechanisms, increasing deployment complexity relative to purely image-based assessment. Image-based systems therefore remain attractive where specialised sensing infrastructure is unavailable, and inference can be performed using cameras on commodity devices.

3 Materials and Methods

The system consists of two clearly separated components. The first is the machine-learning model: a MobileNetV2 CNN used as a frozen feature extractor for binary classification of fresh and rotten produce, trained offline in Python using TensorFlow. The second is the deployment artefact: a static browser-based web application that loads the trained model after conversion to TensorFlow.js and performs inference entirely client-side on a browser-capable device. Throughout the paper, classification quality refers to the model, whereas usability, latency, and offline operation refer to the application. Figure 1 summarises these two stages. The upper pipeline shows offline model development, from the public dataset and preprocessing through MobileNetV2 training and TensorFlow.js conversion. The lower pipeline shows browser-based inference, from image capture or upload and preprocessing to classification, the confidence-derived VFI, indicative storage guidance, and optional PDF report generation.

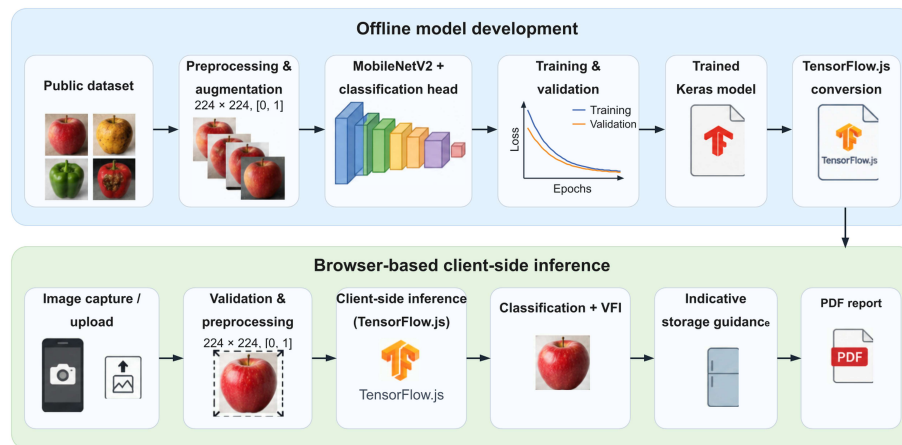


Fig. 1. System architecture showing offline model development and client-side browser inference

3.1 System Design

Figure 2 presents the UML activity diagram for the browser-based inference workflow. The user either captures an image with the device camera or uploads an existing file. The application first validates the input, checking that the file is a supported image format. Invalid input triggers an error message and returns the user to the capture/upload stage. A valid image is resized to 224×224 pixels and rescaled before being passed to the MobileNetV2 model, which extracts visual features and classifies the produce as fresh or rotten, returning an associated confidence score. The classification and confidence-derived VFI are displayed to the user together with indicative storage guidance. The user may then export the results as a PDF report. Figure 3 illustrates the application output presented after inference, including the predicted fresh/rotten class, the associated confidence-derived VFI, and the indicative storage guidance provided to the user.

3.2 Data Preprocessing

The Fresh and Rotten Classification dataset from Kaggle [17] was used, comprising 30,357 labelled images of fresh and rotten fruits and vegetables spanning nine produce types (apples, bananas, bitter melon, capsicum, cucumber, okra, oranges, potatoes, and tomatoes). The dataset ships with its own train/test partition of 23,619 and 6,738 images; carving a 20% validation split from the training partition yields 18,896 training, 4,723 validation, and 6,738 held-out test images. The held-out test partition covers seven of the nine produce types: bitter melon and capsicum appear only in the training partition. This reflects the dataset's own published split, which was retained unchanged rather than re-partitioned so that results remain comparable with other studies using this benchmark; accuracy for these two types therefore cannot be assessed, and the reported test metrics describe performance on the remaining seven. The classes are approximately balanced, with 11,200 fresh and 12,419 rotten images in the training partition and 3,245 fresh and 3,493 rotten images in the test partition, thereby reducing the risk of systematic bias.

For the primary MobileNetV2 model used in the deployed application, all images were resized to 224×224 pixels and pixel values were rescaled to the $[0, 1]$ range by division by 255. To improve generalisation, on-the-fly augmentation was applied during training using random rotations up to 30 degrees, zoom up to 0.2, shear up to 0.2, and random horizontal flips. The same $[0, 1]$ rescaling was used for training, validation, held-out testing, and client-side browser inference, ensuring that the deployed preprocessing pipeline matched that used to evaluate the primary model. We note that the public dataset also contains images produced by offline augmentation.

3.3 Model Architecture and Training

MobileNetV2 was selected as the backbone convolutional architecture for its favourable trade-off between accuracy and computational efficiency in resource-constrained environments. The ImageNet-pretrained convolutional base was frozen for feature extraction, and a compact classification head was appended: a global average pooling layer to aggregate features, a 128-unit dense layer with ReLU activation for representation learning, a dropout layer (rate 0.4) for regularisation, and a single sigmoid-activated output unit for

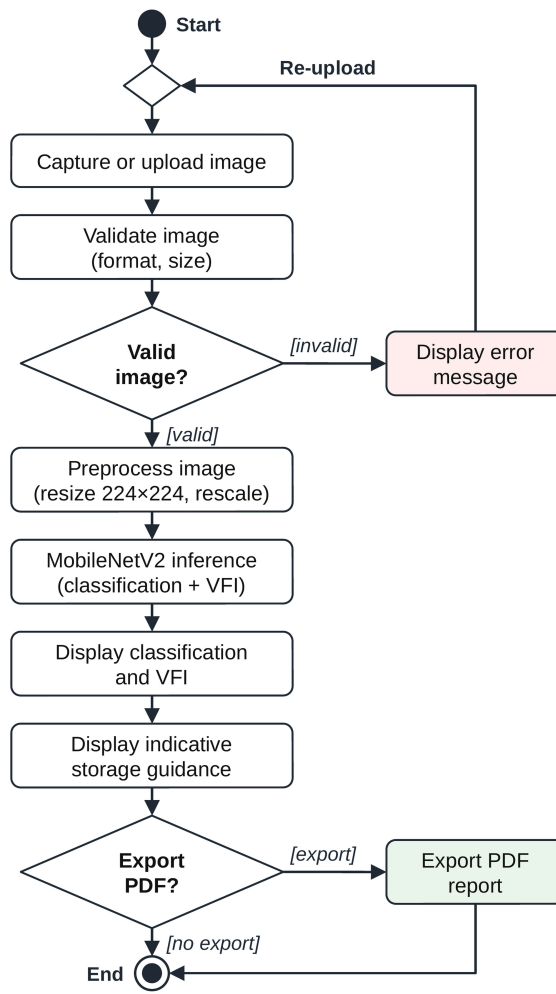


Fig. 2. UML activity diagram of the browser-based inference workflow

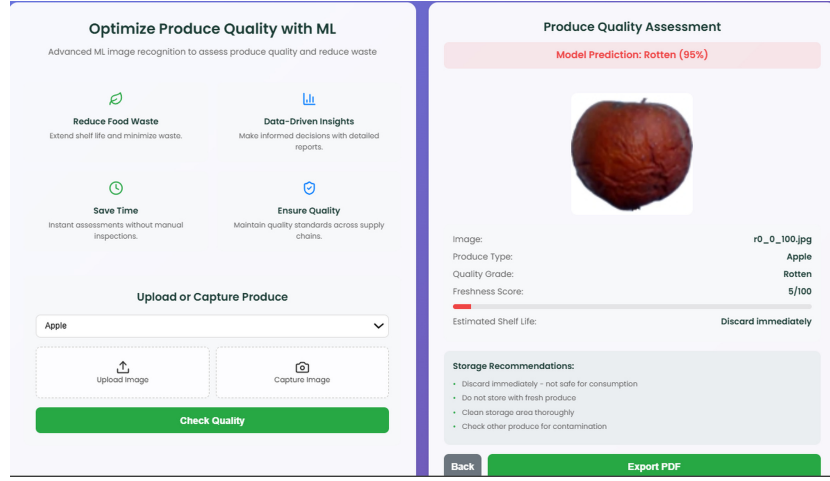


Fig. 3. Application feedback: prediction view showing classification, confidence score, and indicative storage guidance

binary classification. The resulting network has 2,422,081 total parameters, of which 164,097 are trainable, and 2,257,984 are frozen. Table 1 summarises the training configuration; the learning rate, dropout rate, and batch size were selected by a small manual search guided by validation loss.

Table 1. Training configuration

Hyperparameter	Value
Optimiser	Adam
Learning rate	1×10^{-4}
Loss function	Binary cross-entropy
Batch size	32
Maximum epochs	25
Early stopping	Enabled (patience 5, restore best)
Input size	$224 \times 224 \times 3$
Augmentation	rotation 30° , zoom 0.2, shear 0.2, h-flip

The classification head outputs the probability that an image belongs to the rotten class; with fresh encoded as the zero label, the estimated probability of freshness is its complement. The Visual Freshness Index is defined as

$$\text{VFI}(x) = 100 \times \hat{P}(y = \text{fresh} \mid x), \quad (1)$$

where $\hat{P}(y = \text{fresh} \mid x)$ is the model-estimated probability that image x is fresh. The VFI is an interpretable, confidence-based indicator and is evaluated through the calibration and leakage-robustness analyses reported in the Results.

3.4 Baseline Architectures and Preprocessing

To position MobileNetV2 against heavier alternatives, we evaluated ResNet50, EfficientNetB0, and VGG16 in a separate confirmatory architecture-comparison experiment under a matched transfer-learning protocol with a frozen backbone, the same classification head and optimiser, and consistent data splits and augmentation settings. Unlike the primary deployed-model pipeline described in Section 3.2, this comparison supplied each frozen backbone with its architecture-specific native preprocessing function. ResNet50 and VGG16 used Caffe-style mean subtraction, EfficientNetB0 used its corresponding native preprocessing, and MobileNetV2 used its native preprocessing that maps pixel values to the $[-1, 1]$ range. This distinction was necessary because feeding inputs to pretrained frozen backbones outside the distributions expected by their ImageNet preprocessing can substantially degrade the extracted features.

3.5 Evaluation Metrics

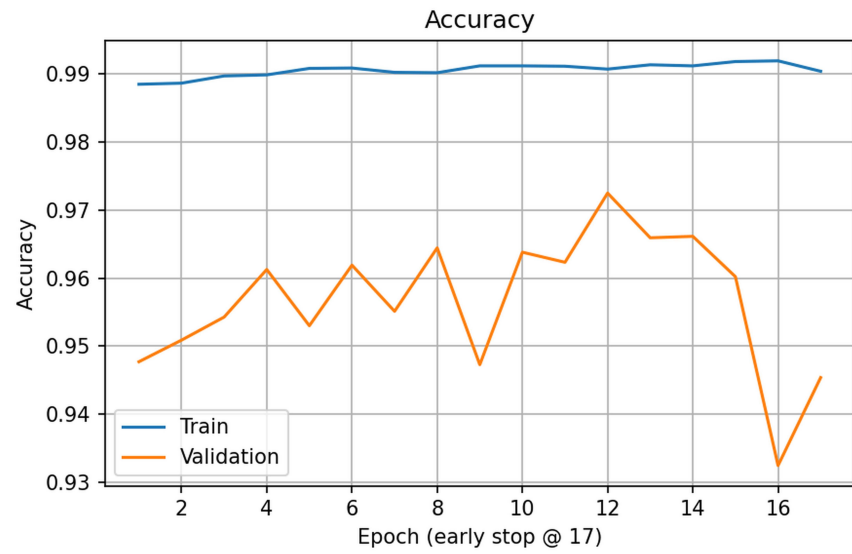
Classification quality is reported with accuracy and per-class precision, recall, and F1-score. Discrimination is reported using the area under the Receiver Operating Characteristic curve (ROC-AUC) and the average precision (AP) from the precision–recall curve, both of which are threshold-independent. Probabilistic quality is reported with the Brier score and the Expected Calibration Error (ECE) computed over ten equal-width confidence bins. Sampling uncertainty in the headline metrics is quantified using 95% bootstrap confidence intervals.

In addition, a dataset-leakage audit is performed: exact byte-duplicate images shared between the train and test partitions are identified by MD5 hashing, and offline-augmented near-duplicates are grouped by a source key formed by stripping class, produce, and augmentation tokens from each filename. The trained model is then re-evaluated, without retraining, on the subset of test images whose identified source never appears in training, providing a source-clean estimate of performance that reduces the influence of identified train–test overlap.

4 Results

4.1 Training Behaviour

The MobileNetV2 model was trained with the configuration summarised in Table 1. Transfer learning with ImageNet-pretrained weights produced rapid convergence: training accuracy exceeded 98% within the first few epochs and reached approximately 98.9%, while validation accuracy stabilised near 96.0%, as shown in Figure 4. Early stopping halted training once validation loss ceased to improve, restoring the best checkpoint. The use of a frozen ImageNet-pretrained backbone and a comparatively small trainable classification head limits the number of parameters updated during training and reduces model complexity [12]. Validation metrics are computed on augmented images, which introduces some noise into the validation curve, whereas the test metrics use clean, unaugmented inputs.



(a)



(b)

Fig. 4. Training behaviour: (a) training and validation accuracy and (b) training and validation loss for the MobileNetV2-based model

4.2 Classification Performance

On the held-out test set of 6,738 images, the MobileNetV2 classifier achieved an overall accuracy of 98.46%. Per-class performance (Table 2) was also balanced: precision and recall were closely matched for both fresh (0.988 and 0.980) and rotten (0.981 and 0.989), giving a macro-averaged F1-score of 0.985. Of the 6,738 test images, 3,179 fresh and 3,455 rotten images were correctly classified, with 66 fresh images misclassified as rotten and 38 rotten images misclassified as fresh.

Table 2. Classification performance on the held-out test set

Class	Precision	Recall	F1-score	Support
Fresh	0.988	0.980	0.984	3,245
Rotten	0.981	0.989	0.985	3,493
Macro avg	0.985	0.985	0.985	6,738
Weighted avg	0.985	0.985	0.985	6,738
Overall accuracy	98.46%			6,738

The model separates the two classes almost perfectly across thresholds: the ROC curve (Figure 5a) gives an AUC of 0.9989, and the precision–recall curve (Figure 5b) gives an average precision of 0.9990. These threshold-independent measures confirm that the strong accuracy is not an artefact of the default decision threshold.

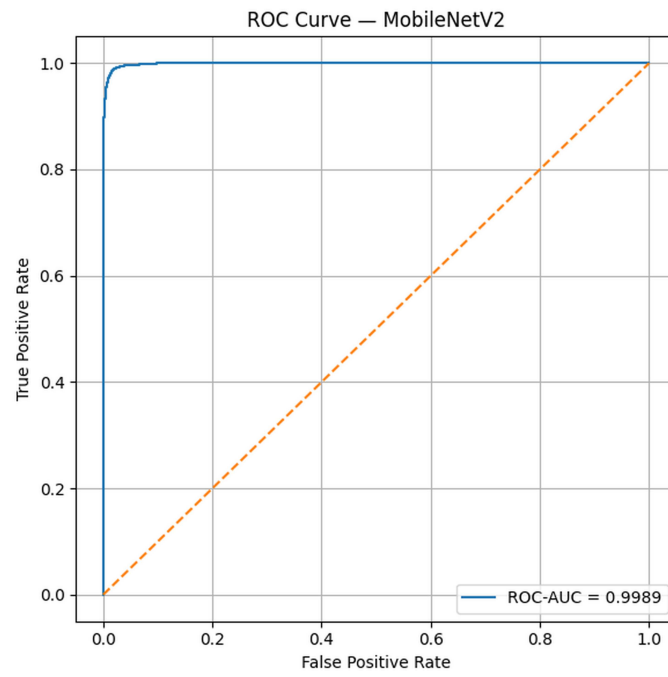
4.3 Calibration and Confidence Intervals

The confidence scores underlying the VFI are well-calibrated; over ten equal-width bins, the ECE is 0.0051, and the Brier score is 0.0122, so the predicted confidence closely tracks empirical accuracy. Bootstrap resampling of the test set (1,000 resamples) gives a 95% confidence interval for accuracy of 98.16%–98.75%, confirming that the headline metrics are stable under test-set sampling.

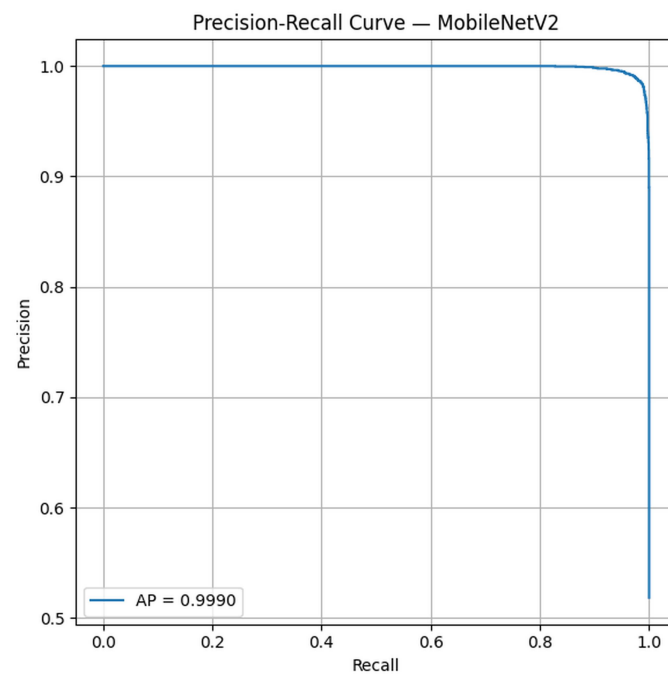
4.4 Baseline Comparison

Table 3 reports the four architectures, each trained for a short, fixed six-epoch schedule under its own native preprocessing. With architecture-specific preprocessing, all models trained successfully, achieving test accuracies between 95.5% and 97.5%. These results confirm the importance of matching the input preprocessing to the requirements of each ImageNet-pretrained backbone.

Under this deliberately short confirmatory schedule, the architectures achieved broadly comparable accuracy. The small differences observed should therefore be interpreted cautiously, as they may partly reflect differences in convergence behaviour under the fixed six-epoch training budget rather than fully converged performance. The deployed MobileNetV2, trained to convergence with early stopping, achieved a held-out test accuracy of 98.46%.



(a)



(b)

Fig. 5. (a) ROC curve and (b) precision–recall curve for the MobileNetV2 classifier

MobileNetV2 was retained as the deployed model not because of a decisive accuracy advantage, but because of its small footprint (2.42M parameters, 9.24 MB) and TensorFlow.js compatibility, which make client-side browser inference practical; ResNet50 and VGG16 contain approximately 10× and 6× more parameters, respectively. A fully converged comparison on a leakage-free, source-disjoint split is left to future work.

Table 3. Architecture comparison under native per-architecture preprocessing on a short six-epoch confirmatory schedule

Model	Total params	Trainable params	Test acc.	F1
EfficientNetB0	4,213,668	164,097	97.49%	0.9757
ResNet50	23,850,113	262,401	96.79%	0.9695
VGG16	14,780,481	65,793	96.10%	0.9626
MobileNetV2	2,422,081	164,097	95.50%	0.9569

4.5 Robustness to Dataset Leakage

The dataset-leakage audit revealed that the public train/test split is heavily contaminated: 4,040 of the 6,738 test images (60.0%) are exact byte-duplicates of training images, and 95.3% share a source image with the training set. To test whether this inflated the reported accuracy, the trained model was re-evaluated, without retraining, on progressively cleaner subsets. Removing exact duplicates left accuracy essentially unchanged at 98.52% (2,698 images), and on the 316 images identified as source-clean, accuracy was 99.37% (95% Wilson confidence interval 97.7%–99.8%) on a class-balanced subset. Accuracy did not decrease after the identified contamination was removed, suggesting that the strong performance is not solely explained by exact or source-related train–test overlap.

5 Discussion

The reported performance is within the range observed in comparable studies. Palakodati et al. [23] reported strong CNN-based transfer-learning performance on a related binary fresh/rotten classification task, while Chuquimarca et al. [7] reported accuracies ranging from approximately 94% to 99% across CNN-based studies of external fruit-quality assessment. In the present study, MobileNetV2 achieved a held-out test accuracy of 98.46%, while accuracy on the source-clean subset was 99.37%. These values fall within the upper range reported in related work, although direct numerical comparison should be interpreted cautiously because the studies use different datasets and evaluation protocols.

The calibration analysis (ECE 0.0051) shows that predicted confidence closely tracks empirical accuracy, which is important because the VFI is presented to users as a graded classification-confidence indicator rather than a bare binary label. The leakage audit

further found that 60% of the benchmark’s held-out test images are exact duplicates of training images. Because this contamination occurs in the public dataset, reported performance on the benchmark may be optimistic when such overlap is not considered. In the present study, accuracy did not decrease after the identified contamination was removed, suggesting that the observed performance was not solely attributable to exact or source-related train–test overlap.

Some produce-quality assessment approaches rely on infrastructure that may be difficult to deploy in resource-constrained settings, including specialised sensing equipment [25,32] and laboratory-based measurements [19]. Muttaqin and Arifudin [16] demonstrated a native mobile approach to fruit-freshness classification, reducing dependence on specialised infrastructure. The browser-based deployment presented here provides a complementary alternative: inference is performed entirely client-side, does not require platform-specific application installation or server-side computation for each prediction, and can operate without continuous connectivity once the required application resources have been loaded. These characteristics reduce deployment requirements and make the approach potentially suitable for users operating browser-capable devices in low-connectivity settings.

Within postharvest supply chains, visual assessment can provide a practical indication of produce condition where more specialised measurements are unavailable. The proposed classifier provides a consistent image-based fresh-versus-rotten assessment, while the confidence-derived VFI communicates the model’s certainty in that classification. Such information may support triage or handling decisions, particularly for cases close to the decision boundary. The scope should nevertheless be interpreted carefully: the system assesses visual freshness at the time of image capture, does not estimate remaining shelf life, and provides storage guidance as heuristic rather than experimentally validated recommendations.

6 Limitations

The study acknowledges several limitations. First, the VFI is the model-estimated probability of the fresh class scaled to a 0–100 range. Although the confidence scores are well calibrated against the binary fresh/rotten labels, this calibration should not be interpreted as validation against physical freshness. The dataset contains no paired physicochemical measurements such as firmness, pH, soluble-solids content, moisture loss, or microbial load, and therefore the relationship between the VFI and these underlying quality attributes remains unknown. A high VFI indicates high model confidence that an image belongs to the visually fresh class, not that the produce has a correspondingly high level of physical or biochemical freshness. Establishing such a relationship would require image-level measurements of relevant quality attributes and an independent validation study. The shelf-life durations and storage suggestions shown by the application are likewise heuristic guidance rather than outputs of the trained model and have not been validated against empirical decay measurements. A further limitation of the leakage analysis in this study is that the source-clean subset contains only 316 images, which increases uncertainty around the corresponding performance estimate, and the source-matching heuristic may fail to identify some near-duplicate relationships.

Further limitations concern robustness and scope. Performance may degrade under low ambient lighting, and changes in viewpoint and illumination alter the colour and surface cues on which the model depends; this is acknowledged qualitatively but not yet quantified through a controlled perturbation study. The system is purely image-based and integrates no temperature, humidity, or gas sensors that would support validated, condition-aware shelf-life estimation. Finally, the baseline architectures were trained under a short six-epoch confirmatory schedule, so the comparison in Table 3 confirms that each architecture trains successfully rather than providing a fully converged benchmark, and a converged, leakage-free comparison is left to future work.

7 Conclusion and Future Work

This study presented a two-component system for fresh produce freshness classification: a MobileNetV2 CNN and a browser-based application that deploys it for offline, client-side inference without server infrastructure or specialised hardware. On a held-out test set of 6,738 images spanning seven produce types, the model achieved 98.46% accuracy (95% CI 98.16%–98.75%), a macro-averaged F1 of 0.985, an ROC-AUC of 0.9989, and an average precision of 0.9990, misclassifying 104 images (1.5%). A calibration analysis of the Visual Freshness Index (ECE 0.0051, Brier 0.0122) confirms that the confidence-derived score is well calibrated against the binary label and suitable as a classification-confidence indicator, while not constituting a physical measurement of freshness.

The primary contribution is the pairing of strong, reliability-characterised classification with a lightweight browser deployment via TensorFlow.js, making the system potentially suitable for resource-constrained and low-connectivity agricultural settings. A leakage audit additionally found that accuracy remained stable after the identified train–test contamination was removed, suggesting that the observed performance was not solely attributable to duplicate or source-related overlap. These findings motivate four directions for future work: (1) multi-class defect detection to capture intermediate quality stages beyond binary fresh/rotten; (2) lighting-robust preprocessing through low-light augmentation or adaptive histogram equalisation, evaluated with a controlled perturbation study; (3) integration of IoT environmental-sensor data for validated, condition-aware shelf-life estimation; and (4) a fully converged, leakage-free architecture benchmark on a source-disjoint split, alongside validation of the VFI against physicochemical freshness measurements. With these extensions, the system could evolve from a binary classifier into a more comprehensive and empirically validated postharvest quality-assessment platform.

Acknowledgments. This work is based on research supported in part by the National Research Foundation of South Africa (Ref. No. SRUG22051310444). The opinions, findings, conclusions, or recommendations expressed are those of the author(s) alone, and the NRF accepts no liability whatsoever in this regard.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alegbeleye, O.O., Singleton, I., Sant'Ana, A.S.: Sources and contamination routes of microbial pathogens to fresh produce during field cultivation: A review. *Food Microbiology* **73**, 177–208 (2018). <https://doi.org/10.1016/j.fm.2018.01.003>
2. Alenazi, M.M., Shafiq, M., Alsadon, A.A., Alhelal, I.M., Alhamdan, A.M., Solieman, T.H.I., Ibrahim, A.A., Shady, M.R., Saad, M.A.O.: Non-destructive assessment of flesh firmness and dietary antioxidants of greenhouse-grown tomato (*solanum lycopersicum* L.) at different fruit maturity stages. *Saudi Journal of Biological Sciences* **27**, 70–90 (2020)
3. Anwar, M.: Crop yield prediction using machine-learning. *Journal of Innovative Computing and Emerging Technologies* (2024), <https://api.semanticscholar.org/CorpusID:273968514>
4. Barz, B., Denzler, J.: Do we train on test data? purging CIFAR of near-duplicates. *Journal of Imaging* **6**(6), 41 (2020). <https://doi.org/10.3390/jimaging6060041>
5. Bibi, M., Rehman, S.U., Mahmood, K., Shoukat, R.S.: An intelligent decision support system for crop yield prediction using machine learning and deep learning algorithms. *Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences* **60**(3), 37–48 (2023). [https://doi.org/10.53560/PPASA\(60-3\)825](https://doi.org/10.53560/PPASA(60-3)825)
6. Binns, C., Low, W.Y.: Vegetable stir fry and fruit salad: The super diet for health. *Asia-Pacific Journal of Public Health* **29**(3), 168–170 (2017)
7. Chuquimarca, L.E., Vintimilla, B.X., Velastin, S.A.: A review of external quality inspection for fruit grading using CNN models. *Artificial Intelligence in Agriculture* **14**, 1–20 (2024)
8. FAO: Global food losses and food waste – extent, causes and prevention (2011)
9. FAO: The state of food and agriculture 2019: Moving forward on food loss and waste reduction (2019), licence: CC BY-NC-SA 3.0 IGO
10. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: *Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 70, pp. 1321–1330 (2017)
11. Kamilaris, A., Prenafeta-Boldú, F.X.: Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture* **147**, 70–90 (2018). <https://doi.org/10.1016/j.compag.2018.02.016>
12. Kornblith, S., Shlens, J., Le, Q.V.: Do better ImageNet models transfer better? In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 2661–2671 (2019)
13. Lamberty, A., Kreyenschmidt, J.: Ambient parameter monitoring in fresh fruit and vegetable supply chains using internet of things-enabled sensor and communication technology. *Foods* **11**, 1777 (2022). <https://doi.org/10.3390/foods11121777>
14. Liakos, K.G., Busato, P., Moshou, D., Pearson, S., Bochtis, D.: Machine learning in agriculture: A review. *Sensors* **18**(8), 2674 (2018). <https://doi.org/10.3390/s18082674>
15. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. *Frontiers in Plant Science* **7**, 1419 (2016). <https://doi.org/10.3389/fpls.2016.01419>
16. Muttaqin, I., Arifudin, R.: Fruit freshness detection using Android-based transfer learning MobileNetV2. *Recursive Journal of Informatics* **2**(1), 8–17 (2024). <https://doi.org/10.15294/rji.v2i1.70845>
17. Nayak, S.S.: Fresh and rotten classification. Kaggle dataset (2026), accessed Mar. 7, 2026. <https://www.kaggle.com/datasets/swoyam2609/fresh-and-stale-classification>
18. Nsunba, P., Isingizwe Nturambirwe, J.F., Nyirenda, C.N.: Streamlining real-time sensor data collection and querying processes from environmental parameters in a food cold store. In:

- 2025 IST-Africa Conference (IST-Africa). pp. 1–11. IEEE (2025). <https://doi.org/10.23919/IST-Africa67297.2025.11060485>
19. Nturambirwe, J.F.I., Hussein, E.A., Vaccari, M., Thron, C., Perold, W.J., Opara, U.L.: Feature reduction for the classification of bruise damage to apple fruit using a contactless FT-NIR spectroscopy with machine learning. *Foods* **12**, 210 (2023). <https://doi.org/10.3390/foods12010210>
 20. Opara, I.K., Fawole, O.A., Opara, U.L.: Postharvest fruit and vegetable wastage and the economic and environmental impacts. In: *Sustainable Approaches in Postharvest Handling and Management*, pp. 439–479. Apple Academic Press (2026)
 21. Opara, I.K., Opara, U.L., Okolie, J.A., Fawole, O.A.: Machine learning application in horticulture and prospects for predicting fresh produce losses and waste: A review. *Plants* **13**, 1200 (2024). <https://doi.org/10.3390/plants13091200>
 22. Opara, U.L., Pathare, P.B.: Bruise damage measurement and analysis of fresh horticultural produce—a review. *Postharvest Biology and Technology* **91**, 9–24 (2014). <https://doi.org/10.1016/j.postharvbio.2013.12.009>
 23. Palakodati, S.S.S., Chirra, V.R.R., Yakobu, D., Bulla, S.: Fresh and rotten fruits classification using CNN and transfer learning. *Revue d'Intelligence Artificielle* **34**(5), 617–622 (2020)
 24. Pedro, A.M.K., Ferreira, M.M.C.: Nondestructive determination of solids and carotenoids in tomato products by near-infrared spectroscopy and multivariate calibration. *Analytical Chemistry* **77**(8), 2505–2511 (2005). <https://doi.org/10.1021/ac048651r>
 25. Rajini, M., Voola, P.: Developing an IoT and ML-driven platform for fruit ripeness evaluation and spoilage detection: A case study on bananas (2024). <https://doi.org/10.2139/ssrn.4838544>, available at SSRN
 26. Rashvand, M., Ren, Y., Sun, D.W., Senge, J., Krupitzer, C., Fadji, T., Miró, M.S., Shenfield, A., Watson, N.J., Zhang, H.: Artificial intelligence for prediction of shelf-life of various food products: Recent advances and ongoing challenges. *Trends in Food Science & Technology* **159**, 104989 (2025). <https://doi.org/10.1016/j.tifs.2025.104989>
 27. Saragih, R., Emanuel, A.: Banana ripeness classification based on deep learning using convolutional neural network. pp. 85–89 (2021). <https://doi.org/10.1109/EIConCIT50028.2021.9431928>
 28. Singh, A., Gupta, R., Kumar, A.: Fresh and rotten fruit detection using deep CNN and MobileNetV2. In: *Advances in IoT and Security with Computational Intelligence*. pp. 219–231. Springer Nature Singapore (2023). https://doi.org/10.1007/978-981-99-5085-0_22
 29. Slavin, J.L., Lloyd, B.: Health benefits of fruits and vegetables. *Advances in Nutrition* **3**(4), 506–516 (2012). <https://doi.org/10.3945/an.112.002154>
 30. Smilkov, D., Thorat, N., Assogba, Y., Yuan, A., Kreeger, N., Yu, P., Zhang, K., Cai, S., Nielsen, E., Soergel, D., Bileschi, S., Terry, M., Nicholson, C., Gupta, S.N., Sirajuddin, S., Sculley, D., Monga, R., Corrado, G., Viégas, F.B., Wattenberg, M.: TensorFlow.js: Machine learning for the web and beyond (2019), <https://arxiv.org/abs/1901.05350>
 31. Srinivasagan, R., Mohammed, M., Alzahrani, A.: TinyML-sensor for shelf life estimation of fresh date fruits. *Sensors* **23**(16), 7081 (2023). <https://doi.org/10.3390/s23167081>
 32. Tace, Y., Tabaa, M., Elfilali, S., Leghris, C., Bensag, H., Renault, E.: Smart irrigation system based on IoT and machine learning. *Energy Reports* **8**, 1025–1036 (2022). <https://doi.org/10.1016/j.egyr.2022.07.088>
 33. World Health Organization: Diet, nutrition and the prevention of chronic diseases: Report of a joint WHO/FAO expert consultation (2003)
 34. Yadav, A., Acharya, A., Visen, A., Patil, A.: Fresh and rotten fruit classification using deep learning. In: *Proceedings of the IJRAR Conference on Recent Trends in Computer Science and Engineering* (2022)