Fresh Scan AI: Deep Learning-Based Rotten Fruit Identification with Gaussian Image Enhancement

Dr. R. Rohini[1], D. Banupriya[2]
[1]*Associate Professor, Department of Computer Science and Engineering, Vivekanandha College of Engineering for Women, Tamilnadu, India*
[2]*PG Scholar, Department of Computer Science and Engineering, Vivekanandha College of Engineering for Women, Tamilnadu, India*

***Abstract*—Fruit that rots after harvest spoils the produce around it and adds to the large share of food lost before it reaches consumers. Sorting by hand is slow, tiring and inconsistent, so automated visual inspection is an attractive alternative. This paper presents Fresh Scan AI, a deep learning system that labels a fruit image as fresh or rotten. The design uses transfer learning on the ResNet50 convolutional neural network, a small Gaussian smoothing filter applied to every input image, and a dropout layer in the classification head. We review the related literature, point out the gaps that matter for practical use, and describe the architecture, preprocessing, training strategy and evaluation plan. The design builds on the benchmark published by Fischer-Brandies et al. [1]; the figures quoted from that study are attributed to it and are not our own measurements. Our experiments are still in progress, so no measured results are reported here.**

***Index Terms*—rotten fruit detection, deep learning, convolutional neural network, ResNet50, transfer learning, Gaussian filter, food waste.**

I. INTRODUCTION

Fruit and vegetables are among the most perishable foods, and a large share of what is grown is lost on the way from farm to consumer. Rot is a major cause, because one decaying fruit releases moisture and microbes that speed up spoilage in its neighbours. Removing rotten pieces early, before storage or packaging, therefore protects the rest of the batch [1]. In most packing houses this inspection is still done by people. Human graders tire and judge differently from one another, and their accuracy falls over a long shift. Computer vision offers a consistent alternative: a camera captures each fruit and a trained model decides whether it is acceptable. Convolutional neural networks (CNNs) suit this job because they learn colour, texture and shape cues, such as discoloration or mould, directly from images instead of relying on hand-written rules [2].

Training a CNN from scratch needs a very large labelled dataset. Transfer learning avoids this by reusing a network pre-trained on ImageNet and retraining only its final layers for the new task [3]. Fischer-Brandies et al. [1] applied this idea to eight fruit types and reported very high accuracy with ResNet50. They also found that mild Gaussian smoothing of the input helped. Fresh Scan AI takes these findings as its starting point.

A. Problem Statement

Given a colour photograph of a single fruit, the system must decide whether the fruit is fresh or rotten, whatever its type. The classifier has to be accurate, must not favour either class, and should keep missed rotten fruit to a minimum, because one undetected rotten fruit can damage a whole stored lot.

B. Objectives and Contributions

This work aims to:

- review the literature on fresh-versus-rotten fruit classification;
- design a ResNet50-based transfer-learning classifier with Gaussian preprocessing;
- define a reproducible training and evaluation protocol on the public dataset of Sultana et al. [4];
- compare ResNet50 with a lighter and a heavier backbone (MobileNetV3Small and VGG16), so that accuracy and computing cost can be weighed together; and

- measure accuracy, balanced accuracy, precision, recall, F1-score, Cohen's kappa and the number of missed rotten fruit.

The main contribution is a clearly specified design and evaluation method that can be reproduced and then extended towards realistic conditions. It does not claim new accuracy figures. Three research questions guide the work. Does mild Gaussian smoothing help a fine-tuned ResNet50 on this task? How much accuracy is lost when a lightweight backbone replaces ResNet50? How well would a model trained on laboratory images carry over to photographs taken in realistic conditions?

C. Scope

We consider binary classification of single fruits photographed against a plain background, covering the eight fruit types in the public dataset: grape, guava, jujube, pomegranate, strawberry, apple, banana and orange. Images containing several fruits, 360-degree inspection, ripeness grading and the mechanical rejection of rotten fruit are outside the scope. This paper covers design and literature only. Implementation results are not yet available.

II. RELATED WORK

A. Machine Learning in Fruit Quality Inspection

Machine learning is used widely in agriculture, and survey studies show that the number of publications has grown quickly over the past decade [5]. Models for fruit quality either predict a numerical indicator, such as firmness, through regression, or assign each sample to a class. This work belongs to the second group and uses only two classes.

B. Classical and Deep Models

Early systems extracted hand-crafted colour and texture features and passed them to a support vector machine (SVM); Bhargava and Bansal used this approach to separate rotten from fresh apples [6]. Karakaya et al. compared several hand-crafted feature sets with features taken from a CNN and found that the CNN features worked best [7]. Wang et al. likewise reported that a ResNet-based network outperformed conventional classifiers on hyperspectral images of blueberries [8]. Together these studies indicate that learned deep features are stronger than hand-designed ones for freshness classification.

C. Image Types

Two kinds of image are common in the literature, RGB and hyperspectral. RGB images are cheap to capture but are sensitive to the background and the lighting. Hyperspectral images are less affected by these factors, but they need costly sensors and much more computing power [2]. Fresh Scan AI uses ordinary RGB images so that it can run with low-cost cameras.

D. Data Size and Variety

Many earlier studies used one or two fruit types or very small image sets, and some tested their model on the same data it was trained on. Such setups say little about robustness [1], [2]. The dataset of Sultana et al. [4] is more demanding. It holds 3,200 images of eight fruits, with 200 fresh and 200 rotten images for each, taken under varied lighting. All images were nevertheless captured in a laboratory, so external validation is still needed.

E. Gaussian Filtering as Preprocessing

A Gaussian filter is a low-pass filter. It suppresses high-frequency noise while keeping edges reasonably well [9]. Fischer-Brandies et al. appear to be the first to use it for fruit classification, and they observed that small kernels helped, whereas large kernels blurred away useful detail [1].

F. Problem Analysis and Research Gap

Table I compares the main approaches and Table II lists the gaps that this work addresses.

TABLE I. COMPARISON OF INSPECTION APPROACHES

Approach	Strength	Limitation	Role
Manual sorting	Flexible human judgement	Slow; fatigue lowers accuracy	Baseline
Hand-crafted features + SVM	Light and simple	Features designed by hand	Classical baseline
CNN from scratch	Learns its own features	Needs very large data	Not used

Approach	Strength	Limitation	Role
Transfer-learned CNN	Works with modest data	Mostly tested on lab images	Core method
+ Gaussian filter	Cheap input regularization	Large kernels blur detail	Preprocessing

TABLE II. RESEARCH GAP ANALYSIS

Gap	Why it matters	Planned response
Small or single-fruit datasets	Weak evidence of robustness	Use eight fruit types and 3,200 images
Testing on training data	Inflates reported accuracy	Keep an untouched test set
Laboratory-only images	May fail on a real conveyor	Add realistic test images later
Accuracy as the only metric	Hides missed rotten fruit	Report recall, F1, kappa and confusion matrix
Heavy models on small devices	Cost may be too high	Compare with MobileNetV3Small

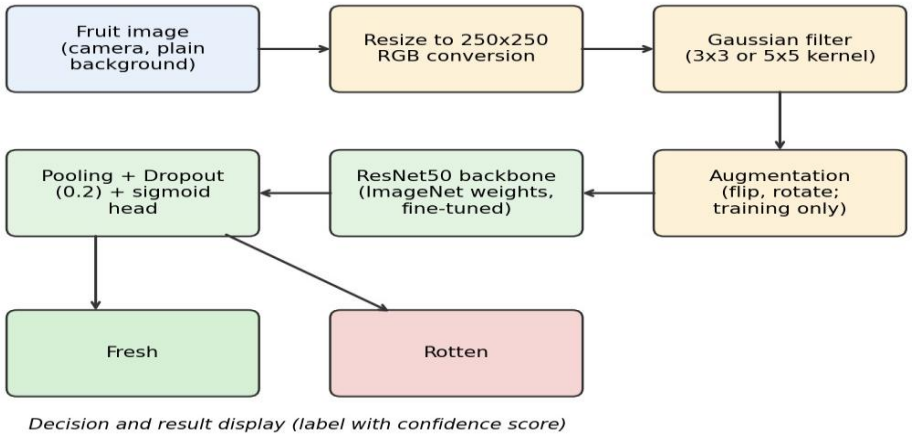

Fig. 1. Fresh Scan AI pipeline: image acquisition, resizing, Gaussian filtering, augmentation (training only), ResNet50 feature extraction, classification head and the fresh or rotten decision.

III. PROPOSED SYSTEM

A. Overview

Fresh Scan AI has five stages: image acquisition, preprocessing, model inference, decision and reporting. Fig. 1 shows the pipeline and Table III lists the components. The figure shows functional modules only; it does not mean that they are already implemented. Each stage will be built and tested on its own before the complete pipeline is joined.

TABLE III. STAGES AND COMPONENTS

Stage	Component	Responsibility
Acquisition	Camera, plain background	Capture one RGB image per fruit
Preprocessing	Resize, Gaussian filter	Standardize and smooth the input
Augmentation	Flip and rotate	Enlarge the training set only

Stage	Component	Responsibility
Model	ResNet50 + custom head	Extract features and classify
Decision	Sigmoid threshold	Output fresh or rotten with a score
Reporting	Result display	Show the label and confidence

B. Design Principles

- Start from pre-trained ImageNet weights.
- Split the data before any augmentation or filtering, so that no test information leaks into training.
- Keep the test set untouched until the final evaluation.
- Treat a missed rotten fruit as costlier than a false alarm.
- Use published figures only as a reference, never as our own results.
- Record software versions and random seeds for repeatability.

C. Assumptions and Limitations

The design assumes one fruit per image on a plain background. A single photograph shows only the camera-facing side, so rot on the far side can be missed; a practical installation would need several views of each fruit. The classifier also only labels fruit. A separate mechanism would be needed to remove the rotten ones from the line.

IV. METHODOLOGY

A. Dataset and Splitting

We use the dataset of Sultana et al. [4], which gives 1,600 rotten and 1,600 fresh images. Each of the 16 classes (eight fruits, two conditions) is split separately into 80% training, 10% validation and 10% test images. This keeps all three sets balanced and gives 2,560 training, 320 validation and 320 test images, the same arrangement as in [1]. Augmentation is applied to the training images only, after the split.

B. Preprocessing

Every image is converted to three-channel RGB and resized to 250 × 250 pixels to reduce computation. The Gaussian filter then replaces each pixel by a weighted average of its neighbours, where the weights follow a bell-shaped curve controlled by the standard deviation σ. For a kernel of size k, the OpenCV rule for σ is:

$$\sigma = 0.3 \times ((k - 1) \times 0.5 - 1) + 0.8 \quad (1)$$

Kernel sizes from 3 × 3 up to 19 × 19 will be tried, each on a fresh copy of the dataset, so that the effect of the amount of smoothing can be seen clearly.

C. Network Architecture

The backbone is ResNet50 with ImageNet weights and without its original top layer. Its residual shortcut connections make very deep networks easier to train [10]. On top of it we place a global average pooling layer, a dropout layer with a rate of 0.2 to limit overfitting, and a single-neuron output layer that gives the probability of the rotten class. VGG16 [11] and MobileNetV3Small [12] are built in the same way, so that the three backbones can be compared fairly.

D. Training Strategy

Training has two stages, following the settings reported in [1]. In the first stage the backbone is frozen and only the new head is trained with the Adam optimizer [13] at a learning rate of 0.0001 for up to 30 epochs. In the second stage the top layers of the backbone are unfrozen and fine-tuned with RMSProp at a learning rate of 0.00001 for up to 40 epochs. In both stages training stops early if the validation accuracy does not improve for four consecutive epochs.

E. Evaluation Metrics

Results will be reported as accuracy, balanced accuracy, true positive rate, true negative rate, positive predictive value, F1-score, Cohen's kappa and a confusion matrix. Because a missed rotten fruit is the more serious error, the confusion matrix will be read with particular care. Each Gaussian setting will be repeated over ten random splits and averaged, so that a single lucky split does not decide the outcome.

F. Module Summary

Table IV summarizes the modules of the pipeline, from the captured image to the displayed label.

TABLE IV. MODULE DESCRIPTION

Module	Input → Output	Processing
Image acquisition	Fruit → RGB image	Capture and check the format

Module	Input → Output	Processing
Preprocessing	RGB image → 250 × 250 smoothed image	Resize and Gaussian filter
Augmentation	Training image → varied copies	Random flip and rotation
Feature extractor	Image → feature map	ResNet50, frozen then fine-tuned
Classification head	Features → score	Pooling, dropout, sigmoid
Reporting	Score → label	Threshold and display

V. PLANNED EVALUATION AND REFERENCE BENCHMARK

Work will proceed in stages: environment setup with Python and TensorFlow/Keras, dataset preparation, baseline training of the three backbones, the Gaussian kernel study, error analysis, and finally tests on additional photographs taken by the authors under more realistic conditions.

Fischer-Brandies et al. [1] report balanced accuracies of 99.38% for VGG16, 99.70% for ResNet50 and 96.71% for MobileNetV3Small on a test set of 320 images. Table V lists their ResNet50 results for selected Gaussian kernel sizes. These published values are the reference against which our reproduction will be compared. They are not our measurements.

TABLE V. RESNET50 BALANCED ACCURACY WITH GAUSSIAN KERNELS, AS REPORTED IN [1]

Kernel size	Balanced accuracy (%)
No filter	99.50
3 × 3	99.66
5 × 5	99.66
7 × 7	99.51
11 × 11	99.31
19 × 19	98.86

The pattern in Table V suggests that mild smoothing helps and heavy smoothing hurts. Our experiments will test whether the same pattern holds. We will treat our reproduction as consistent with [1] only if comparable figures appear under the same protocol, and any difference will be documented and discussed.

VI. CONCLUSION AND FUTURE WORK

This paper described Fresh Scan AI, a transfer-learning design that classifies fruit images as fresh or rotten using a ResNet50 backbone, Gaussian input smoothing and a dropout-regularized head. We reviewed related work, identified the gaps in current evidence and set out a reproducible evaluation plan built on a published benchmark. No new accuracy results are claimed at this stage.

Future work will run the planned experiments and report measured results. It will then test the model on images with leaves, dirt, cluttered backgrounds and uneven lighting, add multiple viewing angles for each fruit, handle images that contain several fruits, and explore explainable predictions. A lightweight version for low-cost edge devices will also be examined.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering, Vivekanandha College of Engineering for Women, for its support.

REFERENCES

[1] L. Fischer-Brandies, L. Müller, J. J. Riegger, and R. Buettner, “Fresh or rotten? Enhancing rotten fruit detection with deep learning and Gaussian filtering,” IEEE Access, vol. 13, pp. 31857–31869, 2025, doi: 10.1109/ACCESS.2025.3542612.

[2] K. Hameed, D. Chai, and A. Rassau, “A comprehensive review of fruit and vegetable classification techniques,” Image Vis. Comput., vol. 80, pp. 24–44, Dec. 2018.

[3] K. R. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” J. Big Data, vol. 3, no. 1, p. 9, May 2016.

[4] N. Sultana, M. Jahan, and M. S. Uddin, “An extensive dataset for successful recognition of fresh and rotten fruits,” Data Brief, vol. 44, Art. no. 108552, Oct. 2022.

[5] L. Benos, A. C. Tagarakis, G. Dolias, R. Berruto, D. Kateris, and D. Bochtis, “Machine learning in

agriculture: A comprehensive updated review," Sensors, vol. 21, no. 11, p. 3758, May 2021.

[6] A. Bhargava and A. Bansal, "Classification and grading of multiple varieties of apple fruit," Food Anal. Methods, vol. 14, no. 7, pp. 1359–1368, Jul. 2021.

[7] D. Karakaya, O. Ulucan, and M. Turkan, "A comparative analysis on fruit freshness classification," in Proc. Innov. Intell. Syst. Appl. Conf. (ASYU), Oct. 2019, pp. 1–4.

[8] Z. Wang, M. Hu, and G. Zhai, "Application of deep learning architectures for accurate and rapid detection of internal mechanical damage of blueberry using hyperspectral transmittance data," Sensors, vol. 18, no. 4, p. 1126, Apr. 2018.

[9] G. Deng and L. W. Cahill, "An adaptive Gaussian filter for noise reduction and edge detection," in Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf., vol. 3, Oct. 1993, pp. 1615–1619.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. 3rd Int. Conf. Learn. Represent., May 2015, pp. 1–14.

[12] A. Howard et al., "Searching for MobileNetV3," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1314–1324.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. 3rd Int. Conf. Learn. Represent., May 2015, pp. 1–15.