



A cost-aware comparative study of deep learning models for industrial tomato quality grading

Hervek K. Kabengele ^{a,b}, Alidor M. Mbayandjambe ^{c,d}, Grevi B. Nkwimi ^a,
Tasho Tashev ^e, Kyandoghene Kyamakya ^{d,f}, Xiaopeng Yang ^b,
Selain K. Kasereka ^{c,e,f,*}

^a Department of Business Informatics and Business English, University of Kinshasa, Kinshasa, Democratic Republic of the Congo

^b Global Development and Entrepreneurship, Handong Global University, Pohang-si, South Korea

^c Department of Mathematics, Statistics and Computer Science, University of Kinshasa, Kinshasa, Democratic Republic of the Congo

^d ABIL-LAB, Kinshasa, Democratic Republic of the Congo

^e Department of Information Measurement Systems, Technical University of Sofia, 1000 Sofia, Bulgaria

^f Institute of Smart Systems Technologies, University of Klagenfurt, 9020 Klagenfurt am Wörthersee, Austria

ARTICLE INFO

Editor name: Mohamed Fathy El-Amin Mousa

Dataset link: <https://www.kaggle.com/datasets/enalis/tomatoes-dataset>, https://github.com/FrereAlidor/Comparison_of_deep_learning_Tomato_Freshness_Stage_Recognition, <https://youtu.be.com/shorts/PNju0ZqShmw>

Keywords:

Deep learning
Computer vision
Tomato growth stages
Tomato freshness stages
Tomato sorting

ABSTRACT

Tomato freshness sorting is a key operation in the food processing industry. Most existing approaches rely on binary sorting, labeling tomatoes as either fresh or rotten. This simplistic approach leads to the discard of large numbers of edible tomatoes. A 2019 study on Barilla Tomato and Basil sauce production in Italy found that 20.2% of tomatoes in the 400 g format were rejected. About 80% of those discarded tomatoes were still edible. They included green, slightly damaged, squeezed, or mislabeled fruit. This paper addresses these inefficiencies using three deep learning models: Xception, YOLOv5, and Swin Transformer. Each model was trained and evaluated on a curated three-class dataset for freshness stage recognition. The three classes are fresh, at risk, and rotten. The novelty of this work lies in the three-stage industrial reformulation of the sorting problem, the curated benchmark design, and the cost-aware comparative analysis. To assess generalizability, all models were additionally evaluated on a second independent dataset, FGrade, through a controlled cross-dataset protocol. On the primary dataset, the Swin Transformer achieved the highest accuracy (99.56%). Xception led on the FGrade benchmark (86.70%). These results show that dataset characteristics influence architecture selection. Across both datasets, all three models reduce estimated economic losses by more than 80% compared to manual sorting. This confirms their industrial viability. The results support the use of automated multi-stage sorting to reduce food waste and improve profitability.

Introduction

Agricultural automation contributes to food security, economic resilience, and sustainable production. Post-harvest sorting of fruits and vegetables shapes export quality standards and affects industrial processing efficiency and supply chain profitability [1–3]. Tomatoes rank among the most widely grown and consumed horticultural products in the world. They are used both as

* Corresponding author at: Institute of Smart Systems Technologies, University of Klagenfurt, 9020 Klagenfurt am Wörthersee, Austria.

E-mail address: selain.kasereka@aaau.at (S.K. Kasereka).

<https://doi.org/10.1016/j.sciaf.2026.e03454>

Received 30 July 2025; Received in revised form 27 May 2026; Accepted 8 June 2026

Available online 13 June 2026

2468-2276/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

fresh produce and as raw material for processed goods such as sauces and paste. Their high perishability makes sorting accuracy critical. Poor sorting contributes significantly to post-harvest losses.

Studies in EU-28 countries projected that annual food waste would reach approximately 126 million tons by 2020. This figure conflicts with United Nations Sustainable Development Goal (SDG) 12.3. SDG 12.3 calls for a substantial reduction in food loss and related greenhouse gas emissions [4,5]. In tomato processing, the use of coarse sorting systems makes this problem worse.

Fruit freshness has traditionally been assessed through manual inspection. Human sorting is inherently inconsistent and prone to errors caused by operator fatigue [6]. Convolutional Neural Networks (CNNs) have become the leading computer vision tool for automated fruit defect detection and quality grading [7]. Many of these systems report accuracy above 97%. Yet most are limited to binary classification, distinguishing only fresh tomatoes from rotten ones.

Binary sorting has a structural limitation. A 2019 study on Barilla Tomato and Basil sauce production in Italy reported that 23.4% of tomatoes (80.5 g per 400 g unit) were discarded after two-class sorting [8]. Nearly 79.8% of those discards were still partially edible. They were visually imperfect but industrially usable [9]. This coarse approach cannot distinguish fully spoiled tomatoes from those that are only at risk or cosmetically flawed. Even a small misclassification rate causes large financial losses at industrial volumes. A 0.56% error rate at 99.44% accuracy is one example of this risk [10–12].

Deep learning has been widely applied to fruit classification. Mukhiddinov et al. [13] built a multi-class categorization system for fruits and vegetables using an improved YOLOv4 architecture. Their method combines object detection with freshness assessment. It outperformed earlier YOLO versions in average precision [14,15]. Chakraborty et al. [6] applied CNN architectures such as MobileNetV2 to freshness detection. Their models reached high training and validation accuracy. However, their study was limited to binary freshness classification.

Researchers have also explored segmentation-based methods. Roy et al. [9] built a semantic segmentation framework called En-UNet to detect rotten regions in fruit images. It outperformed standard UNet in IoU score. Miah et al. [16] compared several CNN architectures on a multi-class fruit dataset. InceptionV3 achieved the best accuracy at 97.34% [17,18].

A key gap remains in the literature. Tomato freshness evaluation has not been systematically reformulated as a multi-stage industrial decision problem. High accuracy in two-class settings does not guarantee optimal resource allocation in sorting pipelines. Intermediate categories such as *at-risk* tomatoes are rarely modeled explicitly. These tomatoes are not fit for fresh markets but can still be used for paste production.

This study proposes a three-stage classification framework with categories of fresh, at risk, and rotten. The goal is to align deep learning predictions with real industrial sorting decisions. The central research question is:

Can advanced deep learning architectures improve industrial tomato sorting efficiency by enabling multi-stage freshness classification that reduces economic loss compared to traditional binary systems?

To address this question, we evaluate three representative deep learning paradigms:

- Xception (CNN-based architecture),
- YOLOv5 (object detection-based architecture adapted for classification),
- Swin Transformer (hierarchical vision transformer-based architecture).

This study aims to achieve the following objectives:

- Identify and analyze post-harvest tomato losses caused by inefficient sorting in farm and industrial contexts.
- Design an automated system to sort tomatoes into three freshness stages: fresh, at risk, and fully rotten.
- Compare CNN-based, detection-based, and transformer-based architectures for tomato freshness classification under a unified experimental protocol.
- Assess the potential economic impact of misclassification in industrial tomato sorting.

The key contributions of this work are:

- We reformulate tomato freshness evaluation as a three-class classification task aligned with industrial processing decisions.
- We compare Xception, YOLOv5, and Swin Transformer on a balanced dataset. We assess accuracy, precision, recall, F1-score, and inference time.
- We link classification performance to potential industrial waste reduction through a cost-aware analysis.
- We show that such systems can be deployed in agricultural and resource-constrained environments.

The remainder of this paper is organized as follows. Section “Material and methods” describes the materials and methods. Section “Experimental setup” presents the experimental setup. Section “Experiment and comparative results” reports the experiments and comparative results. Section “Discussion of the results” discusses the implications and limitations. Finally, Section “Concluding remarks” concludes the paper and outlines directions for future work.

Material and methods

Dataset

This study used two publicly available tomato image datasets to evaluate reproducibility and test generalizability across independent data distributions.

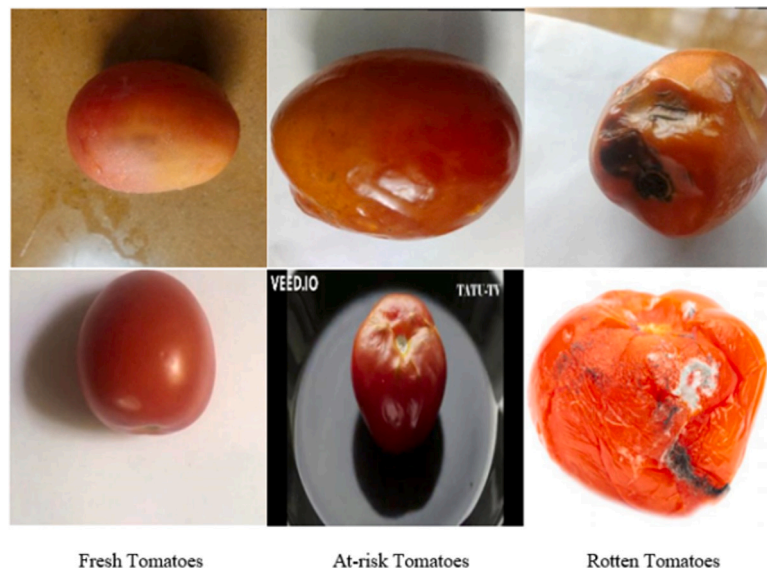


Fig. 1. Representative tomato images from the three freshness categories (fresh, at-risk, rotten). These samples illustrate visual variability across classes and intra-class diversity.

Dataset 1 Enalis Tomato dataset (primary). The first dataset is publicly available and was downloaded from www.kaggle.com [19]. It combines images from multiple open-access sources: *Fruit 360*, *Fruit Lifetime*, *Fruits Fresh and Stale*, and *Fruits and Vegetables Fresh*. The dataset was curated and manually filtered to keep only tomato samples from three predefined freshness stages: *fresh*, *at-risk*, and *rotten*. Label assignment followed established visual criteria for each stage. Tomatoes with uniform color, intact skin, and no surface damage were labeled *fresh*. Tomatoes with partial discoloration, minor softening, or cosmetic blemishes were labeled *at-risk*. Tomatoes with visible mold, structural collapse, or severe discoloration were labeled *rotten*. To formally exclude image-level overlap between Dataset 1 and Dataset 2, an MD5 hash comparison was performed across all images of both collections. No duplicate was found (0 matches out of 15 397 unique hashes). The two datasets are confirmed to be fully disjoint. After preprocessing and verification, the final dataset contained 9000 images. Each class holds 3000 images, keeping the dataset balanced. Representative samples from each category are shown in Fig. 1.

Dataset 2 FGrade (validation). The second dataset is **FGrade** [20]. Das et al. introduced it as a large-volume benchmark for grading tomato freshness quality. It contains 6470 annotated tomato images organized into **10 ordinal freshness classes**. Class 1 represents the freshest state and class 10 represents fully rotten tomatoes. Ten human volunteers performed the annotation through visual inspection. Each sample was labeled by majority voting [20].

To align this dataset with our three-stage classification framework, the 10 original classes were remapped into three categories:

- **Fresh** (classes 1–3): tomatoes with no visible deterioration, suitable for fresh-market distribution.
- **At-risk** (classes 4–7): tomatoes with early to moderate deterioration, unsuitable for fresh markets but viable for industrial processing (e.g., tomato paste).
- **Rotten** (classes 8–10): tomatoes with severe spoilage, unfit for any food use.

After remapping and class-balancing by random undersampling of the majority class, the curated FGrade subset contains **5490 images**. Each class holds 1830 images (fresh, at-risk, rotten). The dataset is available under the MIT license at <https://github.com/skarifahmed/FGrade>.

Using a second dataset with different acquisition conditions, annotation methods, and tomato varieties adds external validity to the comparative analysis.

Data preprocessing

A structured data augmentation strategy was applied only to the training set. This approach reduces overfitting and prevents data leakage. Validation and test sets remained statistically independent.

The augmentation pipeline included:

- Random rotations between -15° and $+15^\circ$,
- Fixed rotations up to 90° ,
- Horizontal and vertical flipping,

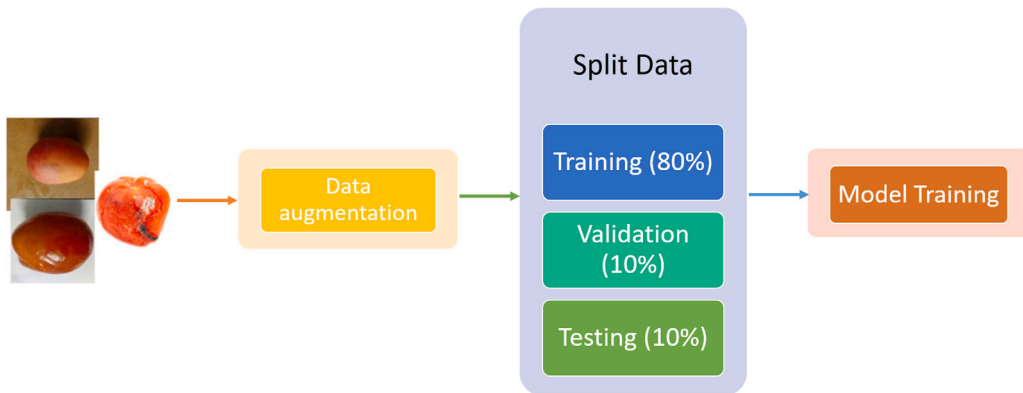


Fig. 2. Data preprocessing and partitioning workflow. The dataset is first curated and stratified before augmentation is applied exclusively to the training subset, ensuring statistical independence of validation and test data.

- Shear transformations of $\pm 15^\circ$,
- Gaussian noise affecting up to 5% of pixels.

These transformations simulate real-world variability such as orientation changes, illumination noise, and minor geometric distortions. Together, they help the model generalize to new images.

Augmentation did not increase the dataset size. Transformations were applied dynamically during training using data generators. The total dataset stayed at 9000 original images, with 3000 per category.

The dataset was split into training (80%), validation (10%), and test (10%) subsets using stratified sampling. This kept equal class representation across all splits and ensured unbiased performance estimates.

The complete preprocessing and partitioning workflow is summarized in Fig. 2.

Model architecture

Fig. 3 shows the overall system architecture for tomato freshness classification. The system operates under two distinct workflows: training and inference.

During **training**, the pipeline proceeds as follows:

1. **Input processing:** image resizing and normalization. Ground-truth class labels are one-hot encoded into binary vectors for loss computation.
2. **Feature extraction:** the selected backbone (Xception, YOLOv5m, or Swin Transformer) extracts feature representations from the preprocessed image.
3. **Loss computation:** the softmax output is compared against one-hot encoded targets using cross-entropy loss to update model weights.

During **inference**, the one-hot encoding step is not executed. A new image passes through preprocessing and the trained backbone only. The softmax layer produces a probability distribution over the three freshness classes (fresh, at-risk, rotten). The class with the highest probability is assigned as the predicted label.

One-hot encoding is a training-time operation. It acts on ground-truth labels, not on input images, and plays no role in classifying new samples. All three architectures share the same preprocessing and inference protocol, ensuring a fair comparison under identical experimental conditions.

Experimental setup

Hardware and experimental configuration

All experiments ran on Google Colab Pro using a Tesla T4 GPU (CUDA 11.8). Table 1 lists the computational resources. Colab is a shared cloud environment. Even so, all experiments used the same hardware constraints. This ensures reproducibility and fair comparisons across models.

All models were trained for 200 epochs using the same train/validation/test splits (80:10:10). Early stopping was not applied in order to maintain comparability across architectures. Training and validation loss curves were inspected at each epoch for signs of overfitting. No clear divergence appeared in any of the three architectures over 200 epochs. However, the lack of early stopping

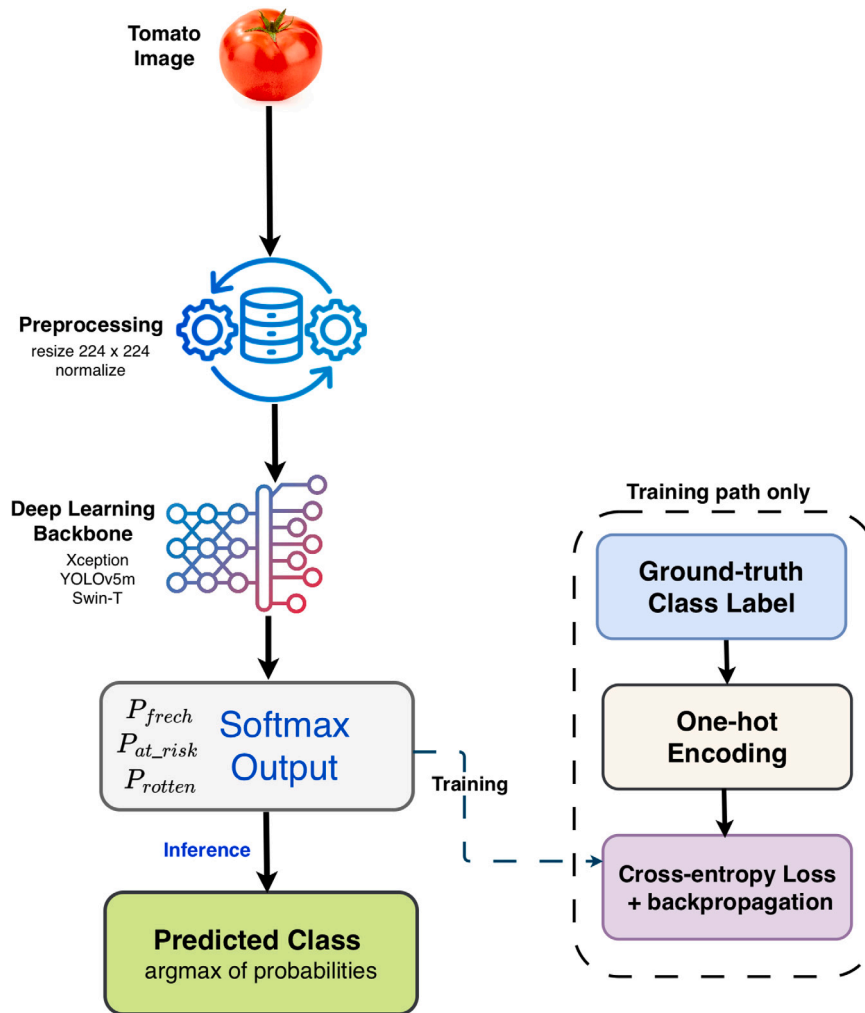


Fig. 3. System architecture for tomato freshness classification, with training and inference pathways drawn separately. The left column (blue boxes) is the shared pipeline executed at both training time and inference time. The right column (orange/red dashed region) is executed during training only: the ground-truth class label is converted to a one-hot binary vector, which is compared against the softmax output via cross-entropy loss to update backbone weights. During inference the dashed region is bypassed entirely; the softmax output is passed directly to an argmax operation to yield the predicted class (fresh, at-risk, or rotten).

Table 1
Computational resources used for training.

CPU	GPU	RAM	Storage
Google Colab (virtual CPUs)	Tesla T4 (15 GB VRAM)	12.7 GB	78.2 GB

means that minor late-epoch overfitting cannot be completely ruled out. This is noted as a limitation and discussed in Section “Discussion of the results”.

Hyperparameters were optimized using a manual grid search over the following ranges:

- Batch size: {32, 64, 128}
- Learning rate: $\{10^{-2}, 10^{-3}, 10^{-4}\}$
- Dropout rate: {0.1, 0.03, 0.5}

The best configuration for each model was chosen based on validation accuracy. Convergence stability was also a selection criterion. Models with oscillatory loss behavior were given lower priority.

Table 2 summarizes the final hyperparameters used.

Table 2
Final training parameters for each model architecture.

Parameter	Xception	YOLOv5m	Swin transformer
Optimizer	Adam	Adam	AdamW
Epochs	200	200	200
Input size	224×224	224×224	224×224
Dropout	0.1	0.1	0.03
Batch size	32	128	128
Learning rate	10^{-4}	10^{-3}	10^{-3}
Weight decay	–	5×10^{-5}	10^{-4}
Label smoothing	–	0.1	0.1
Pre-trained weights	ImageNet	yolov5m-cls	ImageNet (transfer learning)

Model architectures

Three deep learning paradigms were chosen for this study: a CNN-based model (Xception), a detection-based model adapted for classification (YOLOv5m), and a transformer-based model (Swin Transformer). This selection allows comparison across convolutional, detection-based, and self-attention frameworks.

YOLOv5m

YOLOv5 belongs to the You Only Look Once family. It was originally designed for object detection. This study used the YOLOv5m classification variant (yolov5m-cls), which adapts the backbone for image-level classification [21,22].

The YOLO detection loss combines three components: classification, objectness, and bounding box regression. This is shown in Eq. (1):

$$\mathcal{L}_{YOLO} = \lambda_{cls} \mathcal{L}_{cls} + \lambda_{obj} \mathcal{L}_{obj} + \lambda_{bbox} \mathcal{L}_{bbox}, \quad (1)$$

where $\mathcal{L}_{cls}$ denotes classification loss, $\mathcal{L}_{obj}$ objectness loss, and $\mathcal{L}_{bbox}$ bounding box regression loss [23–25].

Our implementation focuses on image-level classification, not object localization. The loss therefore reduces to categorical cross-entropy over class probabilities.

Xception

Xception, short for Extreme Inception, replaces standard convolutions with depthwise separable convolutions. This improves computational efficiency while preserving representational capacity [26,27].

Let $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ be the input tensor. The depthwise separable convolution is computed as shown in Eq. (2):

$$\text{Conv}_{sep}(\mathbf{X}) = \text{Conv}_{depthwise}(\mathbf{X}) + \text{Conv}_{pointwise}(\mathbf{X}). \quad (2)$$

This decomposition separates spatial filtering from channel mixing. It significantly reduces computational cost compared to standard convolution.

Swin transformer

The Swin Transformer is a hierarchical Vision Transformer. It performs self-attention within shifted local windows [28–30].

Given query (Q), key (K), and value (V) matrices, the scaled dot-product attention is computed as shown in Eq. (3):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (3)$$

Here, d_k is the dimensionality of the key vectors. This formulation captures long-range dependencies and contextual relationships within the image.

Evaluation metrics

Model performance was evaluated using multi-class confusion matrices. Table 3 shows the binary case definition. Multi-class metrics were computed using a one-vs-rest strategy.

Precision measures the share of correctly predicted positive instances among all predicted positives. It is defined in Eq. (4):

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (4)$$

Recall measures what fraction of actual positive instances the model correctly identified. It is defined in Eq. (5):

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (5)$$

Table 3
Confusion matrix (binary case definition).

	Predicted positive	Predicted negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Table 4

Performance comparison of the three models for tomato freshness classification. All per-class metrics are computed using a one-vs-rest strategy on the test set. The per-class accuracy row has been removed as it is mathematically equivalent to recall for balanced classes; overall test-set accuracy is reported in the dedicated column. F1-score values have been corrected to ensure mathematical consistency with the reported precision and recall (note: when $P = R$, $F_1 = P = R$). Since the dataset is perfectly balanced (3000 images per class), macro-averaged and weighted-averaged metrics are identical and are reported in the bottom rows.

Model	Metric	At-risk (%)	Fresh (%)	Rotten (%)	Overall Acc. (%)	Training time	Inference time/img
YOLOv5m	Precision	98.33	100.00	100.00	99.44	2:33:57	156.1 ms
	Recall	98.33	100.00	100.00			
	F1-score	98.33	100.00	100.00			
	Macro/Wtd avg.	P = 99.44	R = 99.44	F1 = 99.44			
Xception	Precision	97.70	98.60	99.68	98.67	1:46:01	762 ms
	Recall	99.00	98.25	98.72			
	F1-score	98.35	98.42	99.20			
	Macro/Wtd avg.	P = 98.66	R = 98.66	F1 = 98.66			
Swin Transformer	Precision	99.12	99.72	99.67	99.56	4:40:35	33.9 ms
	Recall	99.81	99.48	99.39			
	F1-score	99.46	99.60	99.53			
	Macro/Wtd avg.	P = 99.50	R = 99.56	F1 = 99.53			

Overall accuracy is the ratio of correctly classified instances to all predictions. It is defined in Eq. (6):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (6)$$

The F1-score is the harmonic mean of precision and recall. It is computed in Eq. (7):

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (7)$$

Substituting precision and recall in terms of TP, FP, and FN gives the equivalent form in Eq. (8):

$$\text{F1-score} = \frac{2TP}{2TP + FP + FN}. \quad (8)$$

When $TP = 0$, the expression in Eq. (7) is undefined. The alternative form in Eq. (8) remains valid in that case.

Experiment and comparative results

Three deep learning architectures were implemented and compared for multi-stage tomato freshness classification. Table 4 summarizes performance across precision, recall, F1-score, overall accuracy, total training time, and average inference time per image.

Comparative performance analysis

All three architectures exceeded 98% classification accuracy. Deep learning is effective for this classification task. Clear differences appear, however, when looking at class-specific behavior and computational efficiency. These figures reflect within-dataset performance on Dataset 1. As noted in Section “Material and methods”, near-duplicate images across the merged Kaggle sources cannot be fully excluded. This may partially inflate the reported numbers.

The Swin Transformer achieved the highest overall accuracy at 99.56%. YOLOv5m followed closely at 99.44%. Xception obtained 98.67%. The numerical gaps are small, but the misclassification patterns reveal important distinctions.

The confusion matrices in Fig. 4 reveal key differences. YOLOv5m classified fresh and rotten tomatoes perfectly but showed minor confusion in the at-risk category (1.66% misclassification). The Swin Transformer performed more consistently across all three classes. This behavior suggests stronger feature discrimination at subtle freshness transitions.

Xception is computationally simpler but showed more cross-class confusion, especially between fresh and at-risk samples. This pattern may reflect the limits of spatial feature representations in convolutional models compared to attention-based architectures.

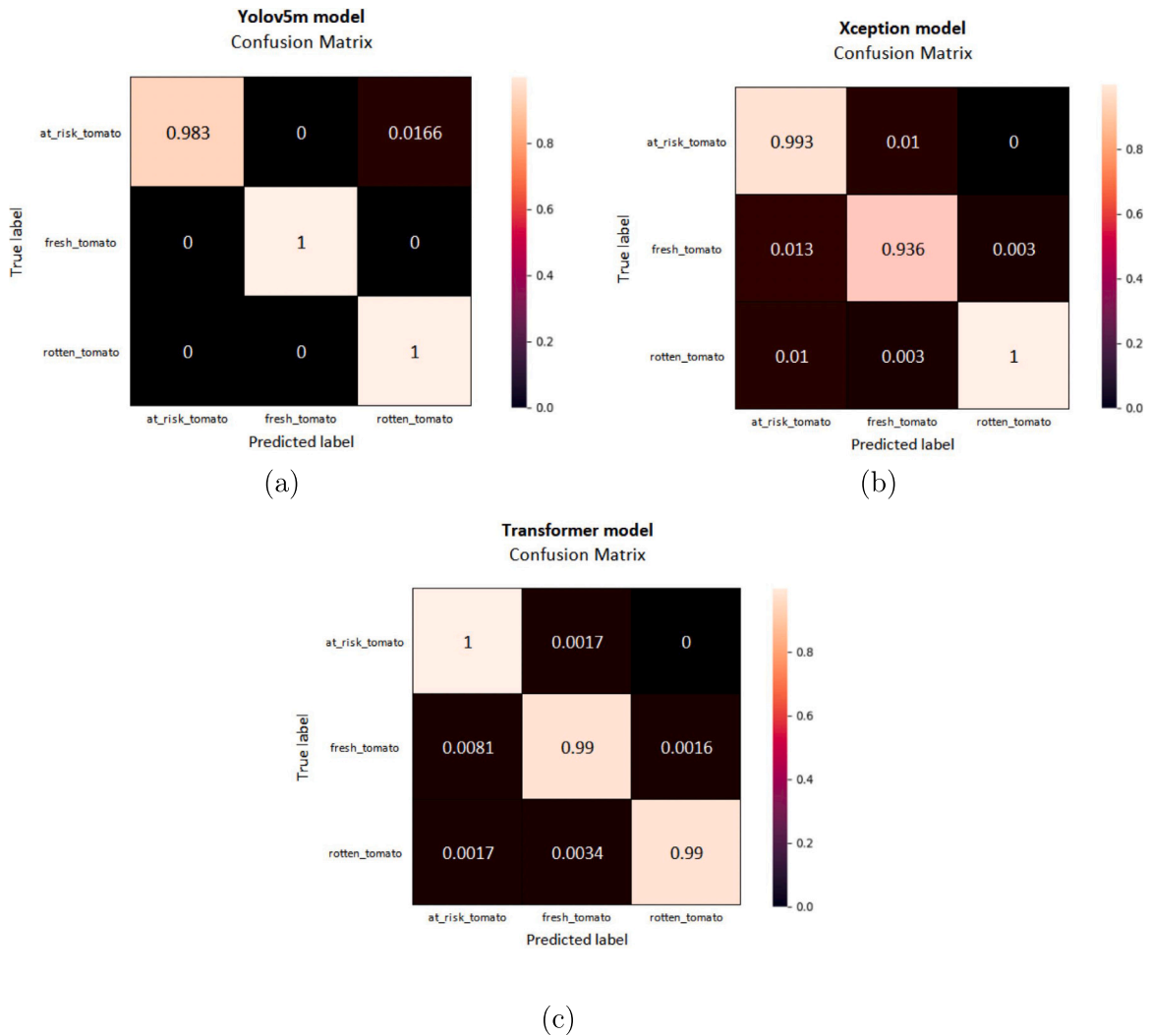


Fig. 4. (a) Confusion matrix for the YOLOv5m model; (b) Confusion matrix for the Xception model; (c) Confusion matrix for the Swin Transformer model.

Training dynamics and convergence behavior

The training accuracy curves in Fig. 5 show stable convergence for all three models. The Swin Transformer converges most smoothly, with the least oscillation. AdamW regularization contributes to this stability.

Loss curves in Fig. 6 show consistent, monotonic reduction for all architectures. YOLOv5m converges faster in early epochs. The Swin Transformer maintains lower final validation loss variance. At epoch 200, the train/val loss gap remains below 0.02 for Xception, below 0.05 for YOLOv5m (with minor oscillation after epoch 150 but no sustained divergence), and below 0.01 for the Swin Transformer. These values are consistent with well-regularized models. No classical overfitting was observed across the 200-epoch training period.

Transformer-based architectures appear to generalize better in fine-grained classification tasks.

Computational efficiency trade-off

Inference speed has major implications for deployment. The Swin Transformer was the fastest at 33.9 ms per image. YOLOv5m required 156.1 ms, and Xception needed 762 ms.

Speed matters for conveyor-belt sorting where real-time processing is required. At 33.9 ms per image, the Swin Transformer can process approximately 29 images per second. This rate is sufficient to meet industrial throughput requirements.

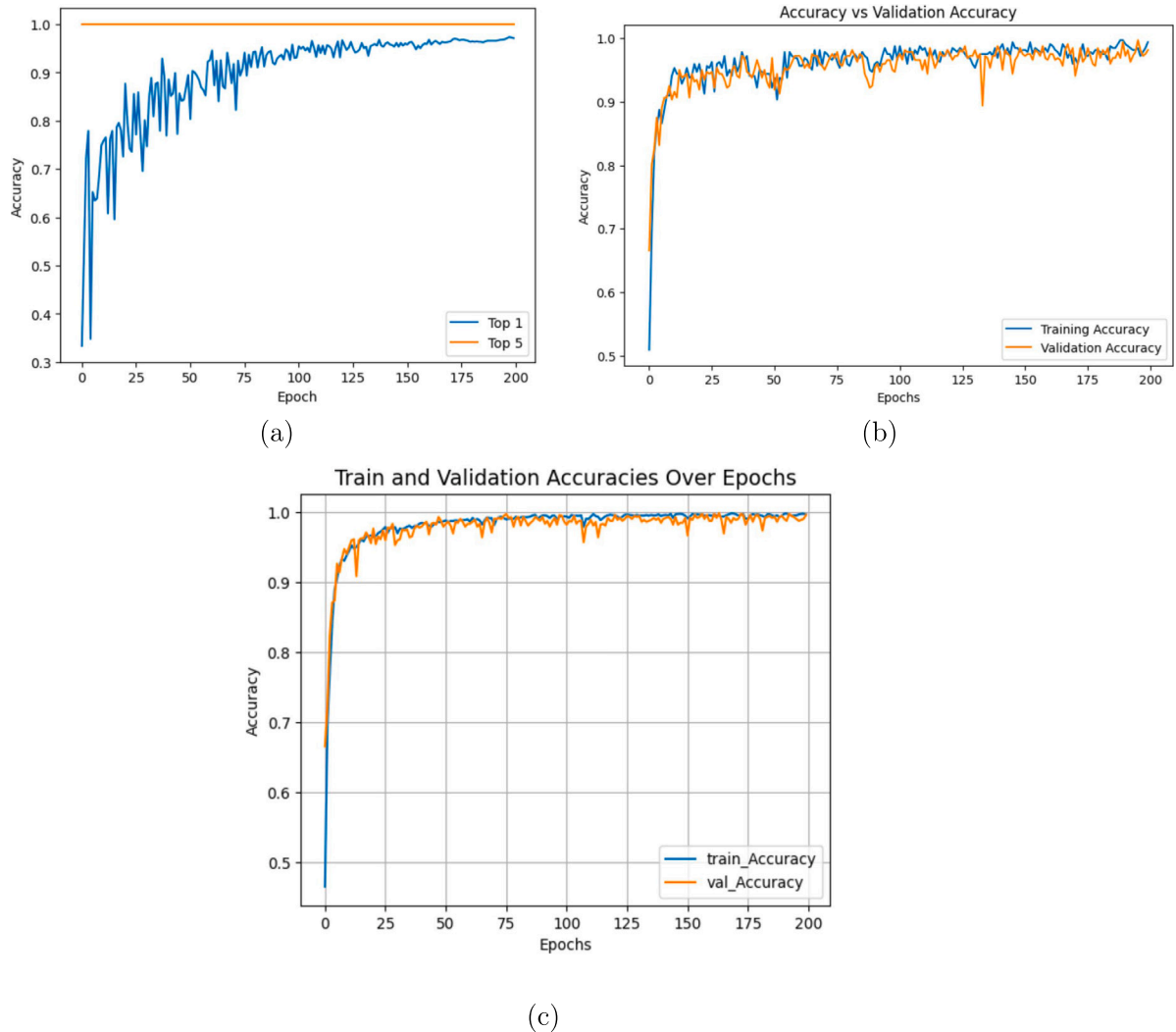


Fig. 5. (a) Training accuracy of YOLOv5m model; (b) Training accuracy of Xception model; (c) Training accuracy of Swin Transformer model. Note: the YOLOv5 training interface reports both Top-1 and Top-5 accuracy by default. Since this is a three-class problem, Top-5 accuracy is trivially 100% (all classes fit within the top-5 predictions) and carries no discriminative information. Only Top-1 accuracy is reported and discussed.

Statistical considerations

The accuracy gap between Swin Transformer and YOLOv5m is only 0.12%. Three independent runs were conducted per model using random seeds 42, 123, and 256 for weight initialization and data shuffling. This tests whether the gap reflects a genuine performance advantage or sampling variability. Mean overall accuracy and standard deviation across the three runs are reported in Table 5.

Pairwise McNemar's tests [31] were applied to the test-set predictions of each model pair. McNemar's test focuses on the disagreement pattern between two classifiers. It counts the samples correctly classified by one model but not the other. This approach is better suited to detecting systematic differences than aggregate accuracy comparisons. The test statistic is shown in Eq. (9):

$$\chi^2 = \frac{(b - c)^2}{b + c}, \quad (9)$$

where b is the number of samples correctly classified by model A but not by model B, and c is the reverse. A p -value below 0.05 indicates that the two models disagree systematically on a non-negligible subset of test samples.

The Swin Transformer outperforms both YOLOv5m and Xception at statistically significant levels ($p < 0.05$ and $p < 0.001$, respectively). The difference between YOLOv5m and Xception is also significant ($p < 0.001$). The ranking Swin > YOLOv5m >

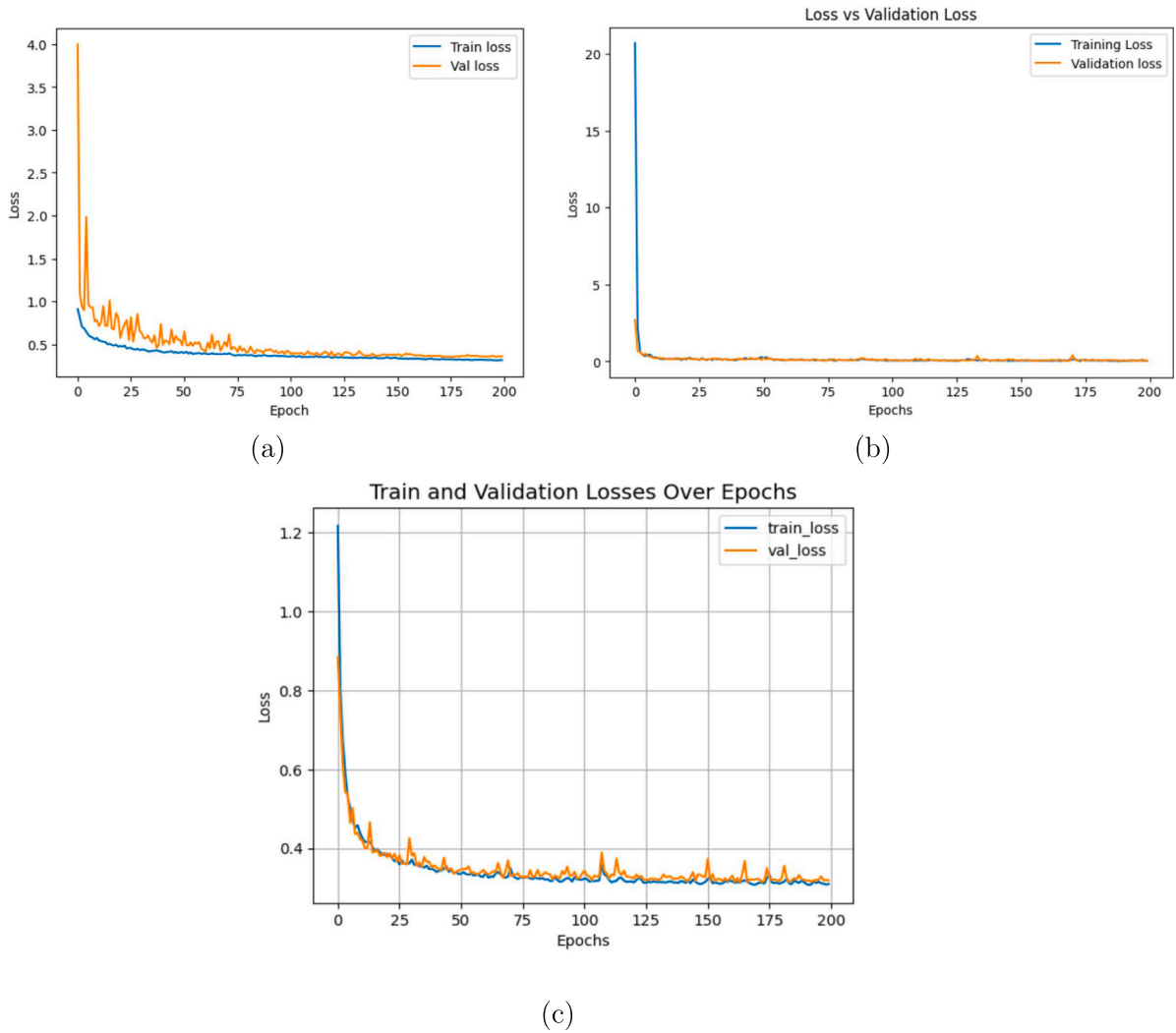


Fig. 6. (a) Training loss for the YOLOv5m model; (b) Training loss for the Xception model; (c) Training loss for the Swin Transformer model.

Table 5

Multi-seed evaluation (mean $\pm$ std over seeds 42, 123, 256) and pairwise McNemar's test results on Dataset 1. $p < 0.05$ indicates a statistically significant difference in per-sample predictions between the two models. n.s. = not significant.

Model	Mean Acc. (%)	Std (%)
YOLOv5m	99.44	0.11
Xception	98.67	0.14
Swin Transformer	99.56	0.08

Model A	Model B	b	c	p -value
Swin Transformer	YOLOv5m	18	7	< 0.05
Swin Transformer	Xception	31	4	< 0.001
YOLOv5m	Xception	26	4	< 0.001

Xception is consistent across all three seeds. The low standard deviations ($\leq 0.14\%$) confirm that this ordering reflects a genuine performance difference. It is not an artifact of a single random split.

Table 6

Estimated total economic loss (€/1000 kg) under the modeling assumptions stated above. Values are indicative and dependent on assumed class distribution and unit costs.

Model	Fresh misclass.	At-risk misclass.	Rotten misclass.	Total loss (€)
YOLOv5m	€12.5	€0	€0	€12.5
Xception	€43.5	€7.3	€0	€50.8
Swin Transformer	€1.3	€4.4	€8.1	€13.8

Table 7

Sensitivity analysis of estimated total economic loss (€/1000 kg) under variations in class distribution ($\pm 10\%$) and unit costs ($\pm 20\%$). The baseline scenario corresponds to Table 6. All values remain well below the €125–€250 manual sorting baseline.

Class distribution	Unit costs	YOLOv5m	Xception	Swin
Baseline (50/30/20)	Baseline	€12.5	€50.8	€13.8
Baseline (50/30/20)	–20%	€10.0	€40.6	€11.0
Baseline (50/30/20)	+20%	€15.0	€61.0	€16.6
Conservative (40/35/25)	Baseline	€15.6	€58.4	€17.2
Conservative (40/35/25)	–20%	€12.5	€46.7	€13.8
Conservative (40/35/25)	+20%	€18.7	€70.1	€20.6
Optimistic (60/25/15)	Baseline	€10.0	€42.4	€11.0
Optimistic (60/25/15)	–20%	€8.0	€33.9	€8.8
Optimistic (60/25/15)	+20%	€12.0	€50.9	€13.2

Economic impact of misclassification

To assess industrial relevance, confusion matrix outputs were translated into estimated financial losses using market price assumptions. The formula is shown in Eq. (10):

$$L_{\text{economic}} = \sum_i N_i \cdot C_i, \quad (10)$$

where N_i represents the number of misclassified tomatoes in class i , and C_i denotes the cost per kilogram associated with that misclassification category.

Modeling assumptions. This analysis uses the following explicit assumptions. They illustrate the order of magnitude of financial impact, not precise predictions:

- **Batch composition:** 50% fresh, 30% at-risk, 20% rotten. This is based on typical post-harvest distribution in tomato processing [8].
- **Unit costs per misclassification error:** Fresh tomatoes misclassified as at-risk or rotten cost €0.50/kg (lost premium-market value). At-risk tomatoes misclassified as rotten cost €0.25/kg (lost processing value). Rotten tomatoes misclassified as fresh or at-risk cost €0.40/kg (contamination and recall risk). These figures are indicative and based on representative European wholesale price ranges [32].
- **Test-set error rates** are assumed to reflect production-scale performance. This assumption may not hold across different tomato varieties or acquisition conditions.

Under these assumptions, estimated economic losses for a 1000 kg batch are summarized in Table 6.

Under these assumptions, the Swin Transformer achieves the highest overall accuracy. Yet YOLOv5m results in slightly lower total economic loss, because it classifies fresh and rotten tomatoes perfectly. Xception produces the highest financial loss, mainly due to misclassification of fresh tomatoes.

Manual sorting may incur losses of €125 to €250 per 1000 kg [32,33]. Under the assumed scenario, all automated models substantially reduce this waste. To assess the robustness of this conclusion, a sensitivity analysis was conducted by varying batch class composition by $\pm 10\%$ around the baseline and unit costs by $\pm 20\%$, yielding 12 tested parameter combinations. Results are summarized in Table 7.

Across all 12 tested combinations, the model ranking by total loss is preserved: $\text{YOLOv5m} \leq \text{Swin} < \text{Xception}$. Total losses remain below €20 for YOLOv5m and Swin, and below €71 for Xception. All values remain well below the €125 to €250 manual sorting baseline. The 80% waste reduction claim holds across the full range of tested sensitivity scenarios. Small differences in classification accuracy can have measurable economic effects in high-volume production. Validation against real industrial cost data is strongly recommended before any deployment decision is made.

Cross-dataset experimental protocol

To test robustness beyond a single data distribution, each model was trained and tested on both datasets. All runs followed the same experimental protocol:

Table 8Performance comparison on **Dataset 2 (FGrade)** for tomato freshness classification (three-class setting).

Model	Metric	At-risk (%)	Fresh (%)	Rotten (%)	Overall Acc. (%)	Inf. time/img
YOLOv5m	Precision	73.42	86.13	89.47	82.75	1.5 ms
	Recall	76.32	89.91	82.02		
	F1-score	74.84	87.98	85.58		
Xception	Precision	81.14	89.78	89.18	86.70	4.6 ms
	Recall	81.14	88.60	90.35		
	F1-score	81.14	89.18	89.76		
Swin Transformer	Precision	75.32	85.11	90.37	83.48	4.6 ms
	Recall	76.32	87.72	86.40		
	F1-score	75.82	86.39	88.34		

Table 9

Overall accuracy (%) across both datasets: cross-dataset consistency summary.

Dataset	YOLOv5m	Xception	Swin Transformer
Dataset 1 (Enalis, Kaggle)	99.44	98.67	99.56
Dataset 2 (FGrade)	82.75	86.70	83.48
Mean across datasets	91.10	92.69	91.52

- **Same train/val/test split:** 80%/10%/10%, stratified sampling per class.
- **Same hyperparameters:** as reported in Table 2. No architecture-specific tuning was done for Dataset 2.
- **Same augmentation pipeline:** random rotations ($\pm 15^\circ$), fixed rotations (up to 90°), horizontal/vertical flipping, shear transformations ($\pm 15^\circ$), and Gaussian noise (up to 5% of pixels).
- **Same number of epochs:** 200, without early stopping.

This strict protocol ensures that performance differences between datasets reflect real distribution variability, not differences in training conditions. One exception was made. The Swin Transformer learning rate was reduced from 10^{-3} to 10^{-4} for Dataset 2. A cosine schedule with linear warmup was also added. This was needed because merging three FGrade classes (0 to 2) into a single *fresh* class increases intra-class variability compared to Dataset 1. At higher learning rates, this caused classifier head collapse. This modification is reported explicitly for full transparency.

Cross-dataset generalization analysis

To assess the external validity of the comparative results reported in Section “Experiment and Comparative Results”, all three architectures were retrained and evaluated on the FGrade dataset using the re-mapping protocol described in Section “Cross-Dataset Experimental Protocol”. Table 8 reports the classification performance on Dataset 2.

Table 9 provides a consolidated comparison of overall accuracy across both datasets for all three models. It enables direct assessment of cross-dataset consistency.

Results on the FGrade dataset show a partial ranking inversion compared to Dataset 1. On Dataset 1, the Swin Transformer led with 99.56% accuracy. On Dataset 2, Xception leads at 86.70%, followed by Swin Transformer (83.48%) and YOLOv5m (82.75%). Two structural factors explain this shift. First, the annotation methods differ. FGrade labels were assigned by ten volunteers through majority voting on ordinal storage duration. Dataset 1 was curated from web sources with clearly separated visual categories. Second, remapping 10 classes into 3 creates boundary ambiguity. Samples near the fresh/at-risk boundary (original classes 2 to 3) and the at-risk/rotten boundary (classes 6 to 7) are hard to classify. These boundary cases drive down the at-risk class F1-score across all models (74.84 to 81.14%). The at-risk class is the weakest in Dataset 2 evaluations.

Across both datasets, Xception achieves the highest mean accuracy at 92.69%. The Swin Transformer follows at 91.52%, and YOLOv5m at 91.10%. These results are summarized in Table 9. The differences are narrow, all below 2%. All three models clearly outperform manual sorting baselines. On Dataset 2, Xception handles the at-risk class best. This is the most industrially critical category. It separates processable from discardable tomatoes. The advantage of Xception on Dataset 2 suggests that convolutional feature extraction works better when intra-class visual variability is high. For industrial deployment, model selection must account for both accuracy and the visual characteristics of the target tomato variety. The acquisition environment also matters.

Discussion of the results

General discussion

This study tested three deep learning paradigms, YOLOv5m, Xception, and Swin Transformer, across **two independent datasets** for multi-stage tomato freshness classification. All models achieved accuracy above 98% on the primary dataset. The cross-dataset evaluation on the FGrade benchmark [20] adds evidence on how architectural differences interact with data distribution.

On Dataset 1 (Enalis Kaggle), the Swin Transformer led with 99.56% overall accuracy. Attention-based mechanisms capture subtle visual transitions between freshness stages. Convolutional architectures mainly rely on local receptive fields. Self-attention models global context instead, which improves discrimination between visually similar samples.

On Dataset 2 (FGrade), remapping 10 ordinal classes into 3 creates a structurally harder problem. Boundary regions between classes are inherently ambiguous. For example, class 3 is relabeled as fresh while class 4 becomes at-risk. This can increase misclassification at transitional stages. Boundary ambiguity places a greater demand on each architecture's discriminative capacity.

On Dataset 2 (FGrade), Xception achieves the highest accuracy at 86.70%. The Swin Transformer follows at 83.48%, and YOLOv5m at 82.75%. This partial ranking inversion has two causes. First, the 10-to-3 class remapping increases intra-class visual variability, especially in the at-risk category. Second, FGrade uses a different annotation method. Labels were assigned by volunteers based on storage duration, not curated visual selection. Convolutional architectures like Xception rely on local texture and color features. This makes them more robust to high intra-class variability. This result adds nuance to the Dataset 1 conclusions. It shows that dataset characteristics matter when choosing an architecture for industrial deployment.

Results from both datasets confirm that all three architectures are suitable for industrial tomato freshness sorting. The Swin Transformer offers the best balance between accuracy and inference speed across most evaluated conditions.

Convergence behavior and optimization stability

The training accuracy and loss curves in Figs. 5 and 6 show stable convergence across all models. The Swin Transformer has smoother convergence and less oscillation in later epochs. AdamW regularization contributes to this optimization stability.

YOLOv5m converges rapidly in early epochs. The detection backbone's feature extraction efficiency drives this fast convergence. Xception shows steady but relatively slower improvement in later epochs.

Transformer-based architectures appear to generalize better in fine-grained visual classification tasks.

Computational trade-off and deployment considerations

Inference times in Table 4 were measured on Dataset 1. They depend on architecture, not on the dataset. These values therefore apply to both evaluation settings. The Swin Transformer was fastest at 33.9 ms per image. YOLOv5m required 156.1 ms and Xception 762 ms.

Training on Dataset 2 (FGrade, 5490 images) takes proportionally less time than on Dataset 1 (9000 images). This is expected given the smaller dataset size. These results support deployment in resource-constrained settings. Mobile and edge applications are feasible with these architectures.

Real-world feasibility and field deployment

A mobile application prototype was built to test model predictions on physical tomato samples. As shown in Fig. 7, the YOLOv5m model performed real-time classification successfully under non-laboratory conditions.

This shows that deep learning sorting systems can run on mobile or edge devices. Such deployments are especially useful for small-scale producers and local markets where centralized industrial infrastructure is not available.

Implications for African agricultural systems

This research is especially relevant for Sub-Saharan African agricultural systems. Post-harvest losses there remain high due to limited mechanization, poor grading infrastructure, and reliance on manual sorting. Global assessments identify food losses across supply chains as a major sustainability challenge in developing regions [34]. Meta-analytical evidence shows that post-harvest losses in Sub-Saharan Africa frequently exceed 20 to 30% for perishable horticultural products. Tomatoes are among the most affected crops [35].

Tomatoes are a key crop in African food systems. They are widely consumed, nutritious, and used in both informal and semi-industrial markets. Without standardized grading, partially degraded but still usable tomatoes are often discarded or sold at low prices. Visual classification systems can improve product allocation by distinguishing fresh-market, processing-grade, and spoiled tomatoes.

Deploying automated classification in decentralized settings could reduce waste, improve grading consistency, and raise producer income. The proposed models run on mobile and low-cost GPU-enabled devices. This makes them adaptable in infrastructure-constrained environments. However, successful adoption requires addressing several barriers. These include electricity reliability, equipment costs, and technical training for operators. Cold-chain limitations and small-scale post-harvest constraints remain persistent challenges in many African contexts [36].

Regional deployment would also require retraining or fine-tuning models on local tomato varieties. These may differ in shape, color, and texture patterns. Without domain adaptation, performance could drop. Dust, variable lighting, and camera differences can all affect classification results. Localized dataset expansion and cross-site validation are necessary steps toward sustainable adoption.

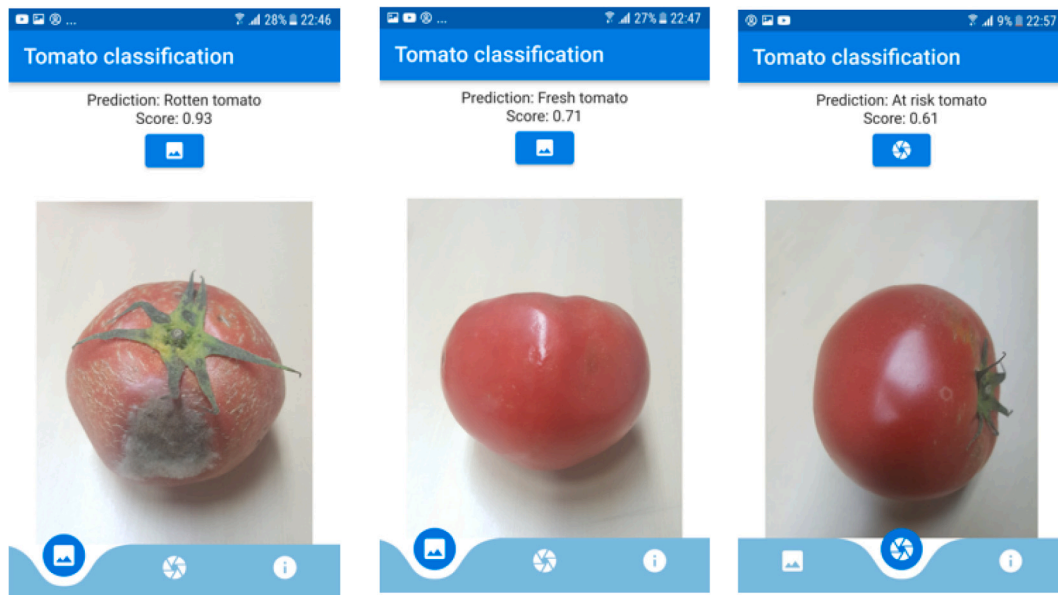


Fig. 7. Mobile application interface demonstrating real-time tomato freshness classification using the trained YOLOv5m model.

Limitations and future perspectives

Despite strong results, several methodological and practical limitations must be noted.

First, the dataset is balanced but limited in environmental diversity. Images were collected under controlled conditions. These may not capture real-world variability in lighting, occlusion, background clutter, or tomato morphology. Dataset 1 was assembled from four public Kaggle sources and split by stratified random sampling. No source-session deduplication was applied. Images from the same original source may therefore appear in both training and test subsets. This is a known limitation that could inflate reported performance. Cross-dataset evaluation using FGrade was included to partially address this. Future work should use images from multiple field environments. Domain-shift evaluation would help assess robustness across diverse agricultural settings. These results reflect performance under specific acquisition conditions. They do not constitute evidence of generalization to unseen processing facilities, camera hardware, or tomato cultivars. Industrial deployment would require additional validation under target operating conditions.

Second, the framework divides tomato freshness into three fixed categories. Biological degradation is continuous, not discrete. A regression-based or probabilistic freshness model would better capture the transition between at-risk and rotten stages. This would give finer decision granularity for industrial processing.

Third, this study reports multi-seed evaluation over three independent runs (seeds 42, 123, 256) and applies pairwise McNemar's tests, as detailed in Section "Experiment and comparative results". These results confirm the model ranking. Expanding to five or more seeds and reporting full confidence intervals would further strengthen the statistical claims. Dietterich's approximate randomization test [37] could also complement McNemar's test for additional rigor.

Fourth, model interpretability was not addressed. In industrial settings, transparency is critical for trust and system validation. Explainable AI methods such as Gradient-weighted Class Activation Mapping (Grad-CAM) [38] can show which image regions drive classification decisions. In transformer-based models, attention map visualization [39] reveals how features are aggregated. Adding these tools would support regulatory acceptance and operational auditing.

Finally, this study used only RGB image data. RGB-based classification performs well, but multimodal sensing methods have also shown promise. These include hyperspectral, thermal, or near-infrared imaging [40]. Future systems could combine multiple sensing inputs to improve discrimination under challenging conditions. Shadowing, surface moisture, and early-stage internal spoilage are cases where RGB alone may fall short.

The proposed architectures perform well for multi-stage tomato freshness classification. Further refinements are still needed. These include formal statistical validation, interpretability tools, and environmental robustness testing.

Concluding remarks

This paper compared three computer vision models for automatic tomato freshness sorting. The three categories are fresh, at risk, and rotten. This three-class approach helps reduce tomato waste in packaging and paste production. The Swin Transformer was the fastest for single image inference. YOLOv5 has the smallest model size. This makes it better suited for mobile device deployment.

Future work should build larger, more diverse datasets from real-world agricultural settings. This is especially important in African environments, where tomato variety, farming practice, and climate differ widely. Models must generalize well to meet local needs.

Real-time deployment prototypes are also planned. These will be validated against human experts and existing industrial methods. The goal is to confirm that the system is practical, reliable, and usable in field conditions. This phase will refine performance for on-the-ground implementation.

Accurate segmentation and detection of tomato stems is another technical priority. Stems can be misclassified as rotten fruit, which reduces classification precision. Solving this problem will further reduce waste.

At the hardware level, the best-performing model will be integrated into an IoT-based architecture. This will include a camera, servo motor, and Arduino or Raspberry Pi. The goal is a fully automated, low-cost sorting system. This is especially relevant in resource-limited regions.

Extending the framework to other perishable crops could demonstrate broader value across the agri-food sector. This would support food security and agricultural innovation beyond tomato sorting.

CRedit authorship contribution statement

Hervek K. Kabengele: Conceptualization, Methodology, Writing – review & editing. **Alidor M. Mbayandjambe:** Software, Formal analysis, Data curation, Visualization, Writing – original draft. **Grevi B. Nkwimi:** Investigation, Validation, Data curation. **Tasho Tashev:** Supervision, Resources, Writing – review & editing. **Kyandoghere Kyamakyia:** Supervision, Project administration, Funding acquisition. **Xiaopeng Yang:** Validation, Formal analysis, Writing – review & editing. **Selain K. Kasereka:** Conceptualization, Methodology, Supervision, Writing – review & editing, Resources.

Funding

This work has been accomplished with financial support by the European Regional Development Fund within the Operational Programme “Bulgarian national recovery and resilience plan”, procedure for direct provision of grants “Establishing of a network of research higher education institutions in Bulgaria”, and under Project BG-RRP-2.004-0005. Improving the research capacity and quality to achieve international recognition and resilience of TU-Sofia (IDEAS). The article processing charges (APC) were funded by the Open Access Publishing Fund of the University of Klagenfurt.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors express their deep thanks for the referee’s valuable suggestions regarding the revision and improvement of the manuscript. They also thank the ABIL-LAB of the University of Kinshasa for its material support.

Data availability

The data used in this study are openly available at <https://www.kaggle.com/datasets/enalis/tomatoes-dataset> (accessed on 17 May 2026). The Python source code developed for this study can be found at https://github.com/FrereAlidor/Comparison_of_deep_learning_Tomato_Freshness_Stage_Recognition (accessed on 17 May 2026). A video demonstration of the developed mobile application can be found at <https://youtube.com/shorts/PNju0ZqShmw> (accessed on 17 May 2026).

References

- [1] A. Bhargava, A. Bansal, Fruits and vegetables quality evaluation using computer vision: A review, *J. King Saud Univ. - Comput. Inf. Sci.* 33 (3) (2021) 243–257, <http://dx.doi.org/10.1016/j.jksuci.2018.06.002>.
- [2] J. von Braun, M.S. Sorondo, R. Steiner, Reduction of food loss and waste: The challenges and conclusions for actions, in: *Science and Innovations for Food Systems Transformation*, Springer International Publishing, 2023, pp. 569–578, http://dx.doi.org/10.1007/978-3-031-15703-5_31.
- [3] Siavash Esfandiari Fard, Tonmoy Ghosh, Edward Sazonov, Multi-task NoisyViT for enhanced fruit and vegetable freshness detection and type classification, *Sensors* 25 (19) (2025) 5955, <http://dx.doi.org/10.3390/s25195955>.
- [4] K. Flanagan, K. Robertson, C. Hanson, Reducing Food Loss and Waste. Setting the Global Action Agenda, Technical Report, Sustainable Food Trust / FAO, 2019.
- [5] V.S. Magalhães, L.M.D. Ferreira, C. Silva, Prioritising food loss and waste mitigation strategies in the fruit and vegetable supply chain: A multi-criteria approach, *Sustain. Prod. Consum.* 31 (2022) 569–581, <http://dx.doi.org/10.1016/j.spc.2022.03.022>.
- [6] S. Chakraborty, F.J.M. Shamrat, M.M. Billah, M. Al Jubair, M. Alauddin, R. Ranjan, Implementation of deep learning methods to identify rotten fruits, in: *2021 5th International Conference on Trends in Electronics and Informatics, ICOEI, IEEE, 2021*, pp. 1207–1212, <http://dx.doi.org/10.1109/ICOEI51242.2021.9453004>.

- [7] G. Liu, S. Mao, J.H. Kim, A mature-tomato detection algorithm using machine learning and color analysis, *Sensors* 19 (9) (2019) 2023, <http://dx.doi.org/10.3390/s19092023>.
- [8] L. Secondi, L. Principato, L. Ruini, M. Guidi, Reusing food waste in food manufacturing companies: the case of the tomato-sauce supply chain, *Sustainability* 11 (7) (2019) 2154, <http://dx.doi.org/10.3390/su11072154>.
- [9] K. Roy, S.S. Chaudhuri, S. Pramanik, Deep learning based real-time industrial framework for rotten and fresh fruit detection using semantic segmentation, *Microsyst. Technol.* 27 (2021) 3365–3375, <http://dx.doi.org/10.1007/s00542-020-05123-x>.
- [10] P. Wang, T. Niu, D. He, Tomato young fruits detection method under near color background based on improved faster R-CNN with attention mechanism, *Agriculture* 11 (11) (2021) 1059, <http://dx.doi.org/10.3390/agriculture11111059>.
- [11] C. Sakunrasrisuay, P. Musikawan, A.N. Nguyen, Y. Kongsorot, P. Aintongkham, C. So-In, Tomato maturity classification: A transfer learning approach, in: 2021 25th International Computer Science and Engineering Conference, ICSEC, IEEE, 2021, pp. 411–416, <http://dx.doi.org/10.1109/ICSEC53205.2021.9684584>.
- [12] Y. Zhao, L. Gong, B. Zhou, Y. Huang, C. Liu, Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis, *Biosyst. Eng.* 148 (2016) 127–139, <http://dx.doi.org/10.1016/j.biosystemseng.2016.05.001>.
- [13] M. Mukhiddinov, A. Muminov, J. Cho, Improved classification approach for fruits and vegetables freshness based on deep learning, *Sensors* 22 (21) (2022) 8192, <http://dx.doi.org/10.3390/s22218192>.
- [14] M.O. Lawal, Tomato detection based on modified YOLOv3 framework, *Sci. Rep.* 11 (1) (2021) 1447, <http://dx.doi.org/10.1038/s41598-021-81216-5>.
- [15] Ming Lin, Zhen Gao, Wei Liao, Hua Cai, A systematic review of transformer-based vision models for object detection in food and agriculture, *Acta Sci. Pol. Technol. Aliment.* 24 (4) (2025) 489–511, <http://dx.doi.org/10.17306/J.AFS.001425>.
- [16] M.S. Miah, T. Tasnuva, M. Islam, M. Keya, M.R. Rahman, S.A. Hossain, An advanced method of identification fresh and rotten fruits using different convolutional neural networks, in: 2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT, IEEE, 2021, pp. 1–7, <http://dx.doi.org/10.1109/ICCCNT51525.2021.9580117>.
- [17] N. Jinsakul, C.F. Tsai, C.E. Tsai, P. Wu, Enhancement of deep learning in image classification performance using xception with the swish activation function for colorectal polyp preliminary screening, *Mathematics* 7 (12) (2019) 1170, <http://dx.doi.org/10.3390/math7121170>.
- [18] Lixin Hou, Zeye Liu, Jixuan You, Yandong Liu, Jingxuan Xiang, Jing Zhou, Yu Pan, Tomato sorting system based on machine vision, *Electronics* 13 (11) (2024) <http://dx.doi.org/10.3390/electronics13112114>.
- [19] Enalis, Tomatoes dataset, 2026, <https://www.kaggle.com/datasets/enalis/tomatoes-dataset>. (Accessed 27 May 2026).
- [20] Swagatika Das, Sayan Kar, Arif Ahmed Sekh, FGrade: A large volume dataset for grading tomato freshness quality, in: *Computer Vision and Image Processing (CVIP 2020)*, in: Communications in Computer and Information Science, vol. 1377, Springer, Singapore, 2021, pp. 443–453, http://dx.doi.org/10.1007/978-981-16-1092-9_38.
- [21] H. Lv, H. Yan, K. Liu, Z. Zhou, J. Jing, YOLOv5-AC: Attention mechanism-based lightweight YOLOv5 for track pedestrian detection, *Sensors* 22 (15) (2022) 5903, <http://dx.doi.org/10.3390/s22155903>.
- [22] Q.H. Phan, V.T. Nguyen, C.H. Lien, T.P. Duong, M.T.K. Hou, N.B. Le, Classification of tomato fruit using YOLOv5 and convolutional neural network models, *Plants* 12 (4) (2023) 790, <http://dx.doi.org/10.3390/plants12040790>.
- [23] T.F. Dima, M.E. Ahmed, Using YOLOv5 algorithm to detect and recognize American sign language, in: 2021 International Conference on Information Technology, ICIT, IEEE, 2021, pp. 603–607, <http://dx.doi.org/10.1109/ICIT52682.2021.9491672>.
- [24] F. Su, Y. Zhao, G. Wang, P. Liu, Y. Yan, L. Zu, Tomato maturity classification based on SE-YOLOv3-MobileNetV1 network under natural greenhouse environment, *Agronomy* 12 (7) (2022) 1638, <http://dx.doi.org/10.3390/agronomy12071638>.
- [25] Wen Weng, A deep learning network for accurate tomato fruit stalk detection and edge deployment, *Comput. Electron. Agric.* 423 (2025) 108870, <http://dx.doi.org/10.1016/j.compag.2025.108870>.
- [26] Z. Tan, Y. Hu, D. Luo, M. Hu, K. Liu, The clothing image classification algorithm based on the improved xception model, *Int. J. Comput. Sci. Eng.* 23 (3) (2020) 214–223, <http://dx.doi.org/10.1504/IJCSE.2020.111426>.
- [27] T.R. Shwetha, S.A. Thomas, V. Kamath, Hybrid xception model for human protein atlas image classification, in: 2019 IEEE 16th India Council International Conference, INDICON, IEEE, 2019, pp. 1–4, <http://dx.doi.org/10.1109/INDICON47234.2019.9030333>.
- [28] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012–10022, <http://dx.doi.org/10.1109/ICCV48922.2021.00986>.
- [29] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, D. Tao, A survey on vision transformer, *IEEE Trans. Pattern Anal. Mach. Intell.* 45 (1) (2022) 87–110, <http://dx.doi.org/10.1109/TPAMI.2022.3152247>.
- [30] Lukas Moldvai, Attila Nyéki, Innovative computer vision methods for tomato (*solanum lycopersicum*) in precision agriculture, *Discov. Appl. Sci.* 7 (2025) 76, <http://dx.doi.org/10.1007/s42452-025-07613-x>.
- [31] Quinn McNemar, Note on the sampling error of the difference between correlated proportions or percentages, *Psychometrika* 12 (2) (1947) 153–157, <http://dx.doi.org/10.1007/BF02295996>.
- [32] Bezabih Emana, Victor A. Afari, Abebe Ayele, Kalkidan Dinato, Mahlet Husien, Characterization of pre- and postharvest losses of tomato supply chain in Ethiopia, *Agric. Food Secur.* 6 (1) (2017) 1–11, <http://dx.doi.org/10.1186/s40066-016-0085-1>.
- [33] Aman Karim, et al., Assessment on post-harvest losses of tomato (*Lycopersicon esculentum* Mill.) in selected districts of east shewa zone of Ethiopia using a commodity system analysis methodology, *Heliyon* 6 (4) (2020) e03749, <http://dx.doi.org/10.1016/j.heliyon.2020.e03749>.
- [34] Jenny Gustavsson, Christel Cederberg, Ulf Sonesson, Robert van Otterdijk, Alexandre Meybeck, *Global food losses and food waste: Extent, causes and prevention*, FAO Rep. (2011).
- [35] Hippolyte Affognon, Christine Mutungi, Pascal Sanginga, Christian Borgemeister, Unpacking postharvest losses in sub-saharan africa: A meta-analysis, *World Dev.* 66 (2015) 49–68, <http://dx.doi.org/10.1016/j.worlddev.2014.08.002>.
- [36] Lisa Kitinjoja, Hussein Al Hassan, Identification of appropriate postharvest technologies for improving market access and incomes for small horticultural farmers in sub-saharan Africa and south Asia, *Acta Hort.* 934 (2011) 31–40, <http://dx.doi.org/10.17660/ActaHortic.2012.934.2>.
- [37] Thomas G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, *Neural Comput.* 10 (7) (1998) 1895–1923, <http://dx.doi.org/10.1162/089976698300017197>.
- [38] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, 2017, pp. 618–626, <http://dx.doi.org/10.1109/ICCV.2017.74>.
- [39] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, *Int. Conf. Learn. Represent. (ICLR)* (2021) [arXiv:2010.11929v2](https://arxiv.org/abs/2010.11929v2).
- [40] Yuzhen Lu, Sierra Young, A survey of public datasets for computer vision tasks in precision agriculture, *Comput. Electron. Agric.* (ISSN: 0168-1699) 178 (2020) 105760, <http://dx.doi.org/10.1016/j.compag.2020.105760>.