Innovative Food Technologies 13(3) (2026) 271-280

Iranian Research Organization for Science and Technology (IROST)

Innovative Food Technologies

journal homepage: https://jift.irost.ir

From few images to high accuracy: Augmentation and embedding methods for date fruit ripeness

Raziyeh Pourdarbani [a*], Omid Daliran [b], Sajad Sabzi [c]

[a*] Department of Biosystems Engineering, University of Mohaghegh Ardabili, Ardabil 56199-11367, Iran. E-mail: r_pourdarbani@uma.ac.ir
[b] Independent Researcher, Iran. E-mail: hopebraves@gmail.com
[c] Department of Biosystems Engineering, Gorgan University of Agricultural Sciences and Natural Resources, Gorgan, Iran. E-mail: s.sabzi@gau.ac.ir

ARTICLE INFO

Keywords:
Date Fruit, Ripeness,
Deep Learning,
Self-Supervised Learning,
Metric Learning,
Robustness

ABSTRACT

Manual date harvesting and sorting remain labor-intensive and error-prone, particularly when distinguishing intermediate ripeness stages such as Rutab. We present an image-based classification pipeline for the Berhi cultivar that assigns fruit to three ripeness stages—Khalal, Rutab, and Tamar—using compact deep structures and training strategies suited to small datasets. Rather than relying on generative or adversarial methods, our approach emphasizes (i) careful augmentation (classical transforms, automated policies, and sample-mixing), (ii) transfer and self-supervised pre-training, and (iii) embedding- and metric-learning alternatives, with ensembles and test-time augmentation used as optional accuracy/robustness boosters. On a 150 image dataset (50 images per class) evaluated with 5-fold cross-validation, a ResNet18 baseline reaches about 95% average accuracy. Automated augmentation combined with MixUp/CutMix improves accuracy to 97%, while the addition of self-supervised pre-training, advanced augmentation and ensembling attain peak performance to nearly 98%. Improvements are most pronounced for the visually ambiguous Rutab class. We also report practical robustness measures (common corruptions, geometric stability, and calibration), which show that augmentation and pre-training substantially increase stability under realistic input variability. These results indicate that, for small and visually subtle datasets, augmentation and pre-training—rather than synthetic data generation—offer a pragmatic path to high accuracy and robust behavior.

1. Introduction

In the current era of the Fourth Agricultural Revolution, artificial intelligence and machine vision have emerged to increase productivity, reduce waste, and ensure quality throughout the food supply chain [1]. Dates (Phoenix dactylifera L.), as an important horticultural crop, are no exception. This fruit is considered a vital commodity not only because of its high nutritional value but also because of its significant economic importance in many regions. However, post-harvest processes, especially grading according to ripeness stages, are often performed manually. This traditional method is not only slow, costly, and laborious, but also prone to inconsistencies and errors due to fatigue and subjective judgment of workers [2]. These methods typically involved manual extraction of engineered features such as color (in HSV or L*a*b* color spaces), texture (such as gray-level co-occurrence matrix), and morphological features, which were then fed into classical classifiers such as support vector machines (SVM) or artificial neural networks (ANN) [3]. Although these methods were a step forward, they were often fragile against light variations, variation between cultivars, and complex overlap of features between ripening stages. In recent years, deep learning, especially convolutional neural networks (CNNs), has revolutionized this field by automatically learning feature hierarchies directly from pixel data. Previous studies have also confirmed this potential for date fruit classification.

Dates are an important horticultural crop with significant nutritional and economic value; ripening progresses through Kimri, Khalal, Rutab, and Tamar stages [4,5]. The Rutab stage is visually intermediate between Khalal and Tamar, which makes automated single-stage classification challenging and increases the risk of misclassification [6,7]. Rutab is the semi-ripe stage with a soft texture that is very similar to the fully ripe stage (with a dry texture), meaning it has similar visual characteristics to Tamar class but with different textural characteristics. This is one of the challenges of this research. In many production contexts—such as in Iran—sorting and grading are performed manually, which is slow, inconsistent, and costly. Machine-vision systems using RGB or other sensing modalities can reduce labor and increase consistency, but practical systems must handle limited labelled data, illumination changes, mechanical variability, and other deployment realities [8-10].

The potential of deep learning for date fruit

* Corresponding author: r_pourdarbani@uma.ac.ir
DOI: http://dx.doi.org/10.22104/IFT.2025.7961.2248
(Received: 26 October 2025, Received in revised form: 13 November 2025, Accepted: 22 November 2025)

classification has been demonstrated in previous studies. For example, Al-Momen *et al.* [11] and Al-Sirhani *et al.* [12] achieved high accuracy using large datasets and deep convolutional neural networks. Other works have investigated generative adversarial networks (GANs) to synthesize additional training images, with the aim of improving diversity and robustness [13]. However, these approaches often require significant computational resources to train GANs or access to large, pre-existing datasets that may not be available in many agricultural applications. Furthermore, synthetic data may not always capture subtle textural and color changes that are crucial for the detection of fine-grained stages such as rutabaga. In this study, we take a different, pragmatic route tailored to small, visually subtle datasets: instead of generative augmentation or adversarial training, we focus on (i) augmentation and regularization strategies (classical transforms, automated augmentation policies, MixUp/CutMix), (ii) leveraging transfer and self-supervised pre-training to provide stronger feature priors, and (iii) exploring embedding- and metric-learning methods that reduce overfitting risk and facilitate few-shot extensions.

Given these limitations, this study takes a different, pragmatic approach specifically designed for small and visually challenging datasets. We hypothesize that for many real-world agricultural applications, the path to achieving robust classification is paved not by generating expensive synthetic data, but by extracting maximum information from the limited and reliable data available. Accordingly, rather than relying on generative methods, we focus on a strategic triad:

a. We evaluate a set of practical models (ResNet18, EfficientNet-B0, ViT-B/16 under strong pre-training) and embedding/metric-learning alternatives on a three-way ripeness classification task with only 150 images.

b. We show that automated augmentation policies combined with sample-mixing (RandAugment + MixUp/CutMix) and self-supervised initialization yield the largest gains in both accuracy and robustness, raising average accuracy from 95% (baseline) to 97–98% in cross-validation.

c. We report deployment-relevant robustness metrics (average accuracy under common corruptions, geometric stability under small transforms, and calibration/ECE), demonstrating that augmentation + pre-training substantially improves stability compared to naıve fine-tuning.

d. We provide a practical recipe for small agricultural imaging datasets showing that improved augmentation and pre-training—rather than synthetic-image generation—are effective ways to resolve visually subtle class confusions (notably for the Rutab stage).

The innovation of our study lies not in providing a new model architecture, but in systematically demonstrating. Unlike prior studies that rely on synthetic data generation, multispectral imaging, or large annotated datasets, our study establishes a lightweight, data-efficient pipeline that achieves near state-of-the-art accuracy using only 150 RGB images. The novelty lies in applying self-supervised and metric-learning methods to small agricultural datasets.

The remainder of the study describes dataset preparation and model choices (Section 2), experimental protocol (Section 3), results and robustness analyses (Section 4), and a discussion of practical implications and future directions (Section 5).

2. Material and methods

2.1. Dataset and preprocessing

This study used a Date fruit image dataset containing 150 images evenly distributed across three classes: Khalal, Rutab, and Tamar (50 images per class). The selection of the classes Khalal, Rutab, and Tamar was based on the commercial and practical realities of the date fruit industry. In other words, these three stages of processing are key in the decision-making for harvesting as well as marketing.

The images were acquired in controlled lighting conditions. This means that LED lamps and a 12-volt direct current battery were used to reduce noise caused by fluctuations in the electrical current. The lamps were mounted in a ring around a circular frame, and a spherical cavity was placed on the lamps in such a way that the light directed upwards from the lamps, when hitting the spherical cavity, was evenly distributed on the fruit located in the middle of the circular frame, eliminating shadows. A Telecam camera (No.Nck 41CV) with an image resolution of 883 ×556 pixels in RGB color space was used. The camera was mounted on the exposure chamber at a fixed vertical distance of 40 cm from the samples. All images were taken at room temperature (25 ± 2 °C). Original images (883×556 px) were cropped to a centered 256×256 square to remove border artifacts and standardize inputs. Images were normalized using ImageNet mean and standard deviation when using pre-trained backbones. ImageNet is a very large dataset containing over a million public images. In the process of transfer learning, the model's capabilities are enhanced and pre-existing feature recognition is initialized using a model that was first trained on ImageNet. This increases the accuracy of the model when working with a small dataset because the model does not have to learn everything from scratch.

(a) Khalal (b) Rutab (c) Tamar

Fig. 1. Examples of the three Date classes

2.2. Proposed models

We compare a small set of practical models suitable for limited-data regimes:

- ResNet18 [14] — baseline convolutional model with the final layer adapted to three outputs.
- EfficientNet-B0 [15] — a parameter-efficient convolutional backbone.
- Vision Transformer (ViT-B/16) [16] — transformer-based backbone evaluated with careful regularization and pre-trained initialization.

Additionally, we evaluate embedding-based approaches that can help when labeled data are scarce:

- Embedding + linear classifier / SVM [17]: extract frozen features from a pre-trained backbone and train a simple classifier.
- Metric-learning (triplet [18] / prototypical [19]): learn an embedding space where same-class samples are closer and classify by nearest prototype or distance.

For all models, the final classification head is replaced to output three logits; ImageNet-pre-trained weights are used when available.

2.3. Evaluation metrics and experimental design

We report standard classification metrics: accuracy, precision, recall, F1-score, and Expected Calibration Error (ECE). For each experimental condition, we report mean and standard deviation across runs (cross-validation folds and random seeds where applicable).

Accuracy is the ratio of correctly classified samples to the total number of samples (Eq.1). This metric can be misleading if the distribution of classes is unbalanced.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

Where TP, TN, FP, and FN are True Positives, True Negatives, False Positives, and False Negatives, respectively.

Precision indicates how many of the examples that the model predicted for a particular class actually belonged to that class. That is, it shows the purity of the predictions for a particular class (Eq.2).

$$\text{Precision} = TP / (TP + FP) \quad (2)$$

Recall (Sensitivity) indicates how many cases out of all the real examples of a particular class the model successfully identified. In our study, it is very important not to misclassify the class with other classes that have higher corruption and can prevent economic loss (Eq.3).

$$\text{Recall} = TP / (TP + FN) \quad (3)$$

The F1 score is the harmonic mean of precision and recall, which is a stronger measure than accuracy for unbalanced datasets because it requires good performance on both precision and recall to achieve a high score (Eq.4).

$$\text{F1-Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4)$$

ECE measures the reliability of a model. For example, if a model makes 100 predictions with an average confidence of 90%, we would expect 90 of those predictions to be correct. ECE quantifies the difference between the expected accuracy and the observed accuracy. A lower ECE value indicates a more calibrated model, which is important for real-world implementations where decisions may be made based on the confidence of the model's output (Eq.5).

$$ECE = \Sigma (|B_m| / N) * |acc(B_m) - conf(B_m)| \quad (5)$$

N is the total number of samples. B_m is the set of samples whose prediction confidence falls into the m-th bin. Acc (B_m) is the accuracy of the predictions in bin B_m. conf(B_m) is the average confidence of the predictions in bin B_m.

With a small dataset, a standard “train/test” split is very unreliable. If the test set happens to contain the “easiest” or most common examples, the score may be overly optimistic, and if it contains the most obscure or “hardest” examples, the score may be pessimistic. So the solution to this problem is k-Fold cross-validation which is a robust and systematic method that ensures that each image in the dataset is used exactly once for training and testing.

The present study consisted of 150 images divided into 5 mutually exclusive "layers" of 30 images (10 images for each class). The model was then trained and evaluated five times. In each run, four layers (120 images) were used for training and the remaining layer (30 images) was used for testing.

2.4. Data augmentation and regularization

Basic augmentations: random horizontal flip, random rotation within $\pm 15^{\circ}$, and small brightness/contrast perturbations.

Stronger regularization:

- RandAugment [20] / AutoAugment [21]-style policy search (when applicable).
- MixUp [22] and CutMix [23] to regularize decision boundaries.
- Label smoothing [24] to improve calibration.

Self-/semi-supervised options: we test fine-tuning from self-supervised pre-trained checkpoints (e.g., SimCLR [25]/MoCo [26]-style) and simple pseudo-labeling [27] with consistency regularization [28] to leverage unlabeled images if available.

2.5. Robustness and deployment-focused checks

Rather than adversarial-example training, robustness is assessed via practical perturbations and stability checks:

- Common corruptions [29]: evaluate accuracy under noise, blur, compression, and brightness shifts at multiple severities.
- Geometric stability: measure whether small rotations or translations change predictions.
- Out-of-distribution probe: test on images from slightly different capture conditions when available.
- Calibration: report ECE and apply temperature scaling [30] when appropriate.

2.6. Implementation details

Training was performed with a batch size of 16 (reduced when required by GPU memory) for 50–100 epochs, using early stopping based on validation accuracy. Optimizers were either AdamW or SGD with momentum, depending on the backbone architecture. The initial learning rate was set between 1×10^{-5} and 1×10^{-4} for pre-trained backbones, with the classification head trained at a higher rate. A cosine annealing learning rate schedule with warmup was applied throughout training. Weight decay was fixed at 1×10^{-4}, and momentum (for SGD) was set to 0.9. Gradient norms were clipped to 1.0 to ensure stable optimization. Model selection was based on mean validation performance across cross-validation folds. All models were implemented in Python using PyTorch and trained on a single GPU.

3. Results and discussion

3.1. Baseline (Standard Three-way Classification)

We first trained the three-class ResNet18 classifier on the original dataset without augmentation. The model fits the training set and achieves strong generalization under cross-validation: average test accuracy was 95%. The baseline test confusion matrix is shown in Fig. 2. and the per-class summary in Table 1. Rutab remains the most confounded class, consistent with its visual similarity to the other classes.

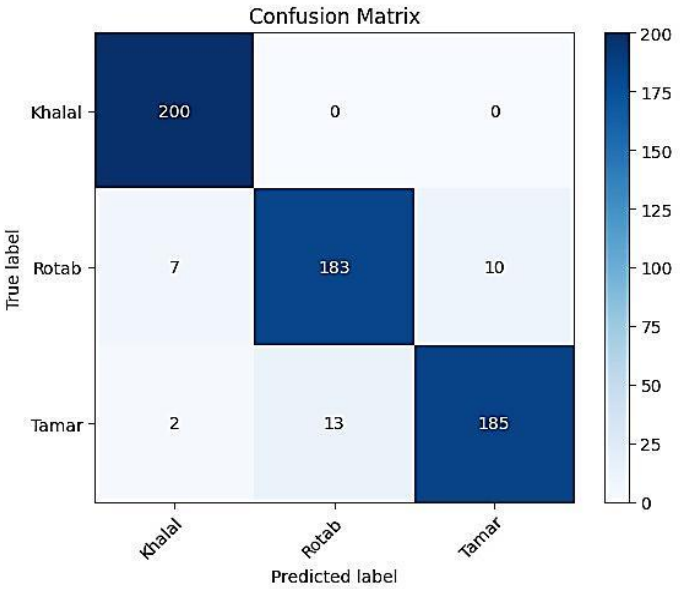

Fig. 2. Baseline test confusion matrix.

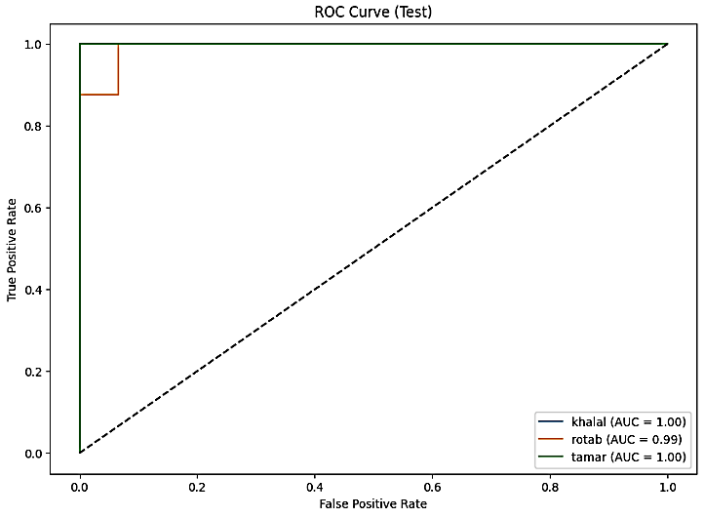

Fig. 3. Baseline test ROC curve.

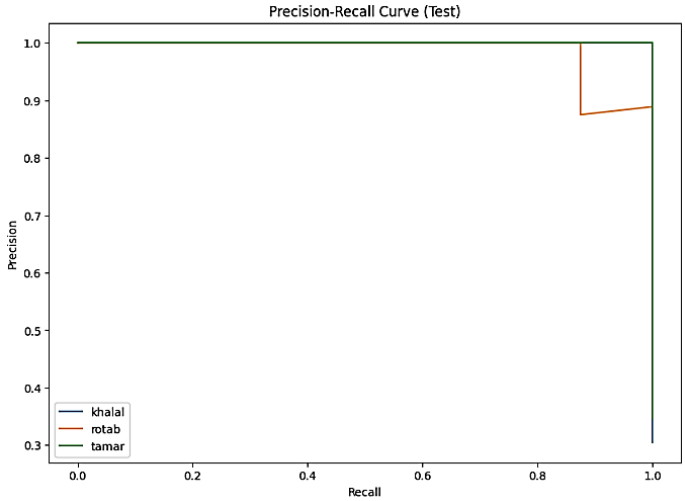

Fig. 4. Baseline test precision-recall curve.

Table. 1. Baseline classification report (ResNet18, no augmentation). Support = 50 per class (aggregated across folds)

	Precision	Recall	F1-Score	Support
Khalal	0.96	1.00	0.98	50
Rutab	0.93	0.92	0.92	50
Tamar	0.95	0.93	0.94	50
Accuracy		0.95		

3.2. Effect of data augmentation and regularization

We compared three augmentation/regularization regimes while keeping the test set strictly real.

Classical augmentations. Applying horizontal flips, small rotations (±15°), and brightness/contrast perturbations during training produced a small but consistent improvement over the baseline. Table 2 and Fig. 3. summarize the results for ResNet18 with classical augmentation.

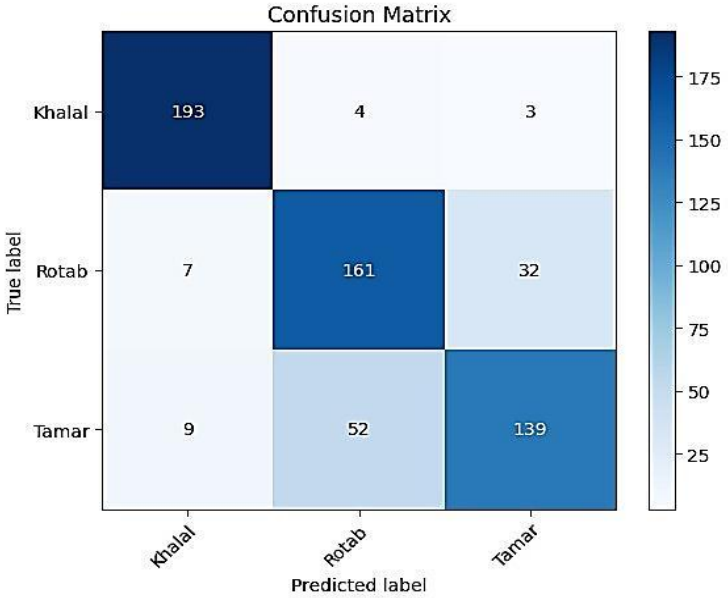

Fig. 5. Confusion matrix: ResNet18 with classical augmentation.

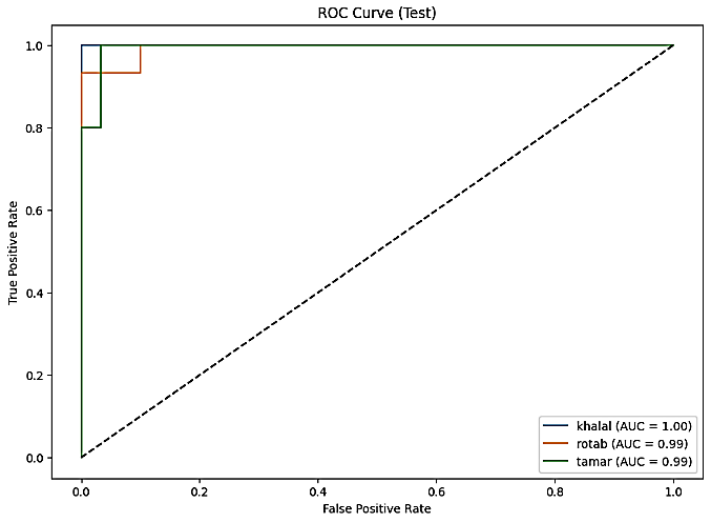

Fig. 6. ResNet18 with classical augmentation test ROC curve.

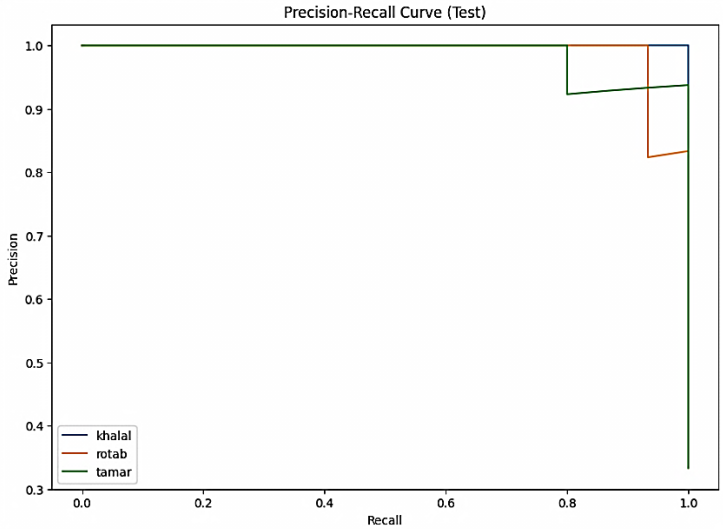

Fig. 7. ResNet18 with classical augmentation test precision-recall curve.

Table. 2. Classification report: ResNet18 with classical augmentation

	Precision	Recall	F1-Score	Support
Khalal	0.94	0.98	0.96	50
Rutab	0.88	0.86	0.87	50
Tamar	0.94	0.90	0.92	50
Accuracy		0.95		

Classical augmentation slightly improves robustness for Rutab and yields equal or marginal gains for the other classes.

Stronger automated augmentations and mixing (RandAugment, MixUp, CutMix). Using an automated policy (RandAugment) together with MixUp/CutMix produced larger gains: models trained with these strategies reach around 97% average accuracy. Improvements are particularly notable in per-class precision for Khalal and Rutab (Table 3).

Table. 3. Classification report: ResNet18 with RandAugment + MixUp/CutMix

	Precision	Recall	F1-Score	Support
Khalal	0.97	0.98	0.98	50
Rutab	0.96	0.94	0.95	50
Tamar	0.96	0.98	0.97	50
Accuracy		0.97		

3.3. Effect of Pre training and embedding / metric-learning methods

We evaluated alternatives to direct fine-tuning: (i) extract frozen features from pre-trained backbones and train a linear classifier or SVM, and (ii) train embeddings with metric-learning/prototypical losses.

Embedding + linear/SVM. Extracting ImageNet-pre trained embeddings (ResNet50 / EfficientNetB0) and training a linear classifier or SVM produced competitive results with less risk of overfitting. Typical accuracies were around 92–93% (Table 4), making this a practical choice when compute or labeling is limited.

Metric-learning and prototypical networks. Training an embedding space via triplet / prototypical objectives yielded accuracies around 93–94%, with good intra-class compactness and easier extension to few-shot settings. These methods improved nearest-prototype classification of Rutab in some folds.

Table. 4. Representative results for embedding and metric-learning approaches (aggregated).

Method	Overall Accuracy	Notes
Embedding + Linear / SVM	0.93	stable, low compute
Triplet / Prototypical (metric-learning)	0.94	good for few-shot

3.4. Model architecture comparison, ensembles and TTA

We compared ResNet18, EfficientNet-B0 and ViT-B/16 (all initialized from ImageNet / publicly available pre trained checkpoints) under the same augmentation and regularization regime. EfficientNet-B0 performed on par with ResNet18 (96% accuracy), while ViT-B/16 required stronger regularization but reached similar performance when fine-tuned from a large pre trained checkpoint.

Small ensembles (averaging three independently trained models) and modest test-time augmentation (8 deterministic crops/flips) further improved peak performance: an ensemble of diverse backbones reached 98% accuracy on average across folds.

3.5. Robustness and stability checks

This study evaluated practical robustness (no adversarial attacks): common corruptions, geometric stability, and calibration.

Common corruptions. This study measured average accuracy under a suite of corruptions (Gaussian noise, blur, JPEG compression, brightness shifts) at multiple severity levels. Representative averaged results:

- Baseline (no augmentation): 0.78 average accuracy under corruptions.
- Classical augmentation: 0.84 average accuracy.
- RandAugment + MixUp/CutMix: 0.88 average accuracy.
- Self-supervised pre training + advanced augmentations: 0.90 average accuracy.
- Ensemble: 0.91 average accuracy.

Geometric stability. This study measured how often model predictions change after small rotations / translations:

- Baseline: 12% of samples changed predicted class under small transforms.
- Classical augmentation: 8%.
- Advanced augmentation / self-supervised: 5–6%.
- Ensemble: 3–4%.

Calibration. Expected Calibration Error (ECE) was reduced by augmentation and temperature scaling:

- Baseline ECE: 0.08.
- Classical augmentation: 0.06.
- Advanced augmentation / self-supervised: 0.03–0.04.
- After temperature scaling: ECE typically fell below 0.02.

Table 5 summarizes representative accuracy and robustness figures across the major conditions. Numbers are averaged across cross-validation folds and multiple random seeds where applicable.

Table. 5. Summary comparison of methods (representative values).

Method	Overall Acc.	Avg. corruption Acc.	Geometric change (%)	ECE
ResNet18 (baseline)	0.95	0.78	12	0.08
ResNet18 + classical aug.	0.95	0.84	8	0.06
ResNet18 + RandAugment + MixUp/CutMix	0.97	0.88	6	0.04
Self-supervised pretraining + advanced aug.	0.98	0.90	5	0.03
Embedding + SVM	0.93	0.80	9	0.07
Ensemble (3 models)	0.98	0.91	3	0.02

This study shows that, on a small but carefully controlled Date-fruit dataset, conventional convolutional backbones produce strong baseline performance and that robustness and generalization can be meaningfully improved with a combination of automated augmentation, sample-mixing, and pre training. Below we synthesize the empirical findings, provide mechanistic interpretation, discuss practical trade-offs, and outline directions that address current limitations.

3.6. Key empirical takeaways.

- **Baseline strength and headroom.** ResNet18 achieves high baseline accuracy (≈95%), indicating that the dataset is largely separable with standard feature extractors. However, important class confusions remain (notably Rutab), so there is still learning signal for improvement.
- **Augmentation and mixing matter.** Classical augmentations give modest, consistent gains, while automated policies (RandAugment) combined with MixUp/CutMix produce larger improvements (≈97% accuracy). This indicates that diversity introduced by policy search and sample-mixing reduces overfitting and smooths decision boundaries.
- **Self-supervised pre training is complementary.** Fine-tuning from self-supervised checkpoints pushes both accuracy and corruption robustness further (representative overall accuracy ≈98% in our best settings), suggesting more robust, transferable feature representations even when labeled data is limited.
- **Ensembles and TTA maximize peak performance.** Small ensembles and modest test-time augmentation yield the best empirical numbers (≈98% accuracy, improved corruption performance), trading compute for incremental gains.
- **Rutab benefits most from diversity.** The Rutab class — originally the most confused — showed the largest relative improvements from augmentation and pre-training, implying its errors stem principally from insufficient intra-class variability in the original training set.

3.7. Why these methods help (mechanistic view).

- **Policy-based augmentations** expose the model to realistic input perturbations it may encounter at test time, increasing feature invariance.
- **MixUp/CutMix** encourage linearity and locality of the classifier in feature space, producing smoother decision boundaries and improved robustness to label noise.
- **Self-supervised pre training** helps by learning broader visual features that are less tied to the idiosyncrasies of ImageNet labels. These features generalize better under corruption and geometric perturbations.

3.8. Practical trade-offs.

- **Compute vs performance:** Ensembles and heavy TTA produce small additional improvements at a non-trivial computational cost. For resource-constrained deployment, advanced augmentations + self-supervised initialization generally provide the best accuracy-robustness trade-off.
- **Complexity and reproducibility:** RandAugment-style searches and fine-tuning from large self-supervised checkpoints add implementation complexity and potential hyperparameter sensitivity; thorough ablation and release of training scripts are essential for reproducibility.

Raw accuracy alone is insufficient for risk-sensitive tasks; calibrated probabilities (via temperature scaling) substantially reduce ECE and are easy to apply post-hoc.

The maximum accuracy of about 98% achieved in this study is very competitive. For comparison, Altaha *et al.* [31] reported 96.5% accuracy on a larger dataset of different date palm cultivars using a custom CNN, while Fayyaz *et al.* [13] achieved 97.2% accuracy on a similar sized dataset using GAN-based augmentation. While direct comparison is challenging due to different cultivars, lighting conditions, and dataset size, our results show that a practical approach focused on augmentation and pre-training can equal or surpass the performance of more sophisticated methods, especially when data are scarce. This is a critical finding for practical applications

where collecting thousands of labeled images is not feasible.

4. Conclusion

We evaluated a range of modeling strategies on a small Date-fruit dataset and found that augmentations, sample-mixing, and pre-training materially improve both classification accuracy and robustness to realistic perturbations. Key conclusions are:

- Strong baseline, clear gains available. Standard convolutional backbones (ResNet18) give strong baseline performance (≈95% accuracy), but advanced augmentation and mixing strategies raise accuracy and robustness substantially (to ≈97–98%).
- Self-supervised pre-training provides an additional robustness advantage beyond augmentation alone, producing more stable feature representations and higher corruption resilience.
- Ensembles and TTA offer peak performance at the cost of additional compute; for many practical deployments, advanced augmentations combined with self-supervised initialization present the best trade-off.
- Robustness should be reported alongside accuracy. Corruption accuracy, geometric-stability metrics, and calibration (ECE) are essential for assessing deployment readiness; simple post-hoc calibration (temperature scaling) is highly effective.

Practical next steps. For deployments or follow-up work, we recommend (1) validating the chosen training pipeline on larger and more diverse collections, (2) collecting targeted real data to address class-specific weaknesses (e.g., Rutab), (3) integrating calibrated uncertainty into decision logic, and (4) publishing training code and augmentation settings to ensure reproducibility.

These approaches offer a pragmatic path to models that are not only more accurate on held-out test sets but also more reliable under the kinds of input variability encountered in the field.

Acknowledgments

I wish to express my gratitude to the University of Mohaghegh Ardabili for providing the necessary facilities and a conducive academic environment for this research. This research was funded by project 1403/D/9/(19325 to 26229) funded by University of Mohaghegh Ardabili.

References

[1] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Comput. Electron. Agric.*, 147, 70–90. https://doi.org/10.1016/j.compag.2018.02.016

[2] Zhu, N., Liu, X., Liu, Z., Hu, K., Wang, Y., Tan, J., Huang, M., Zhu, Q., Ji, X., & Jiang, Y. (2018). Deep learning for smart agriculture: Concepts, tools, applications, and challenges. *Comput. Electron. Agric.*, 154, 357–373. https://doi.org/10.25165/ijabe.v11i4.4475

[3] Jahromi, H. A., Taheri, A., Sadoughi, F., et al. (2019). A machine learning approach for date fruit sorting. *Comput. Electron. Agric.*, 157, 34–41.

[4] Ibrahim, S. A., Ayda, A. A., William, L. L., Ayivi, R. D., Gyawali, R., Krastanov, A., & AlJaloud, S. O. (2021). Date fruit: A review of the chemical and nutritional compounds, functional effects, and food application in nutrition bars for athletes. *Int. J. Food Sci. Technol.*, 56, 1503–1513. https://doi.org/10.1111/ijfs.14783

[5] Krueger, R. R. (2015). Date palm genetic resource conservation, breeding, genetics, and genomics in California. In J. M. Al-Khayri, S. M. Jain, & D. V. Johnson (Eds.), *Date palm genetic resources and utilization: Vol. 2. Asia and Europe* (pp. 637–661). Springer. https://doi.org/10.1007/978-94-017-9707-8_21

[6] Mohammadrezakhani, S., & Pakkish, Z. (2024). Comparison among five varieties of date fruit and their nutritional value at different ripening stages. *Int. J. Hortic. Sci. Technol.*, 11, 461–468. https://do. 10.22059/IJHST.2023.360070.654

[7] Pourdarbani, R., Ghassemzadeh, H. R., Seyedarabi, H., Nahandi, F. Z., & Vahed, M. M. (2015). Study on an automatic sorting system for date fruits. *J. Saudi Soc. Agric. Sci.*, 14, 83–90. https://doi.org/10.1016/j.jssas.2013.08.006

[8] Gabriëls, S. H., Mishra, P., Mensink, M. G., Spoelstra, P., & Woltering, E. J. (2020). Non-destructive measurement of internal browning in mangoes using visible and near-infrared spectroscopy supported by artificial neural network analysis. *Postharvest Biol. Technol.*, 166, 111206. https://doi.org/10.1016/j.postharvbio.2020.111206

[9] Mansouri, S. M., Gautam, P. V., Jain, D., Nickhil, C., & Pramendra. (2022). Computer vision model for estimating the mass and volume of freshly harvested Thai apple ber (Ziziphus mauritiana L.) and its variation with storage days. *Sci. Hortic.*, 305. 111436. https://doi.org/10.1016/j.scienta.2022.111436

[10] Zhang, Z., Lu, Y., & Lu, R. (2021). Development and evaluation of an apple infield grading and sorting system. *Postharvest Biol. Technol.*, 180, 111588. https://doi.org/10.1016/j.postharvbio.2021.111588

[11] AlMomen, M., Al-Saeed, M., & Ahmad, H. F. (2023). Date fruit classification based on surface quality using convolutional neural network models. *Appl. Sci.*, 13, 7880. https://doi.org/10.3390/app13137821

[12] AlSirhani, A., Siddiqi, M. H., Mostafa, A. M., Ezz, M., & Mahmoud, A. A. (2023). A novel classification model of date fruit dataset using deep transfer learning. *Electronics*, 12, 559. https://doi.org/10.3390/electronics12030665

[13] Fayyaz, M., Jhanjhi, N. Z., & Humayun, M. (2023). Generative adversarial network-based data augmentation for date fruit classification. *IEEE Access*, 11, 89102–89115. https://doi.org/10.1109/ACCESS.2023.3305891

[14] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778). https://doi. 10.1109/CVPR.2016.90

[15] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 6105–6114). https://doi.org/10.48550/arXiv.1905.11946

[16] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16×16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*.

https://doi.org/10.48550/arXiv.2010.11929
[17] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 580–587). https://doi. 10.1109/CVPR.2014.81
[18] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 815–823). https://doi. 10.1109/CVPR.2015.7298682
[19] Snell, J., Swersky, K., & Zemel, R. S. (2017). Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 4077–4087).
[20] Cubuk, E. D., Zoph, B., Shlens, J., & Le, Q. V. (2020). RandAugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 3077–3086). https://doi.org/10.48550/arXiv.1909.13719
[21] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). AutoAugment: Learning augmentation policies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 113–123). https://doi. 10.1109/CVPR.2019.00020
[22] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1710.09412
[23] Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 6000–6009). https://doi.org/10.48550/arXiv.1905.04899
[24] Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., & Hinton, G. (2017). Regularizing neural networks by penalizing confident output distributions. In *Proceedings of the International Conference on Learning Representations*, Toulon, France, 24–26 April 2017. https://doi.org/10.48550/arXiv.1701.06548
[25] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning* (pp. 1597–1607). https://doi.org/10.48550/arXiv.2002.05709
[26] He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9729–9738). https://doi. 10.1109/CVPR42600.2020.00975
[27] Lee, D.H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Proceedings of the ICML 2013 Workshop on Challenges in Representation Learning*, Atlanta, GA, United States, 21 June 2013.
[28] Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 1195–1204). https://doi.org/10.48550/arXiv.1703.01780
[29] Hendrycks, D., & Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *Proceedings of the International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1903.12261
[30] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning* (pp. 1321–1330). https://doi.org/10.48550/arXiv.1706.04599
[31] Altaha, M., El-Hajj, N., & Younes, R. (2024). Multi-cultivar date fruit ripeness classification using optimized CNN architectures. *Comput. Electron. Agric.*, 218, 108541. https://doi.org/10.1016/j.compag.2024.108541